'68' MICRO JOURNAL

$2.95USA

Australia	A $4.75	New Zealand	NZ $ 6.50
Singapore	S $9.45	Hong Kong	H $23.50
Malaysia	M $9.45	Sweden	30:-SEK

VOLUME VI ISSUE X • Devoted to the 68XX User • October 1984

"Small Computers Doing Big Things"

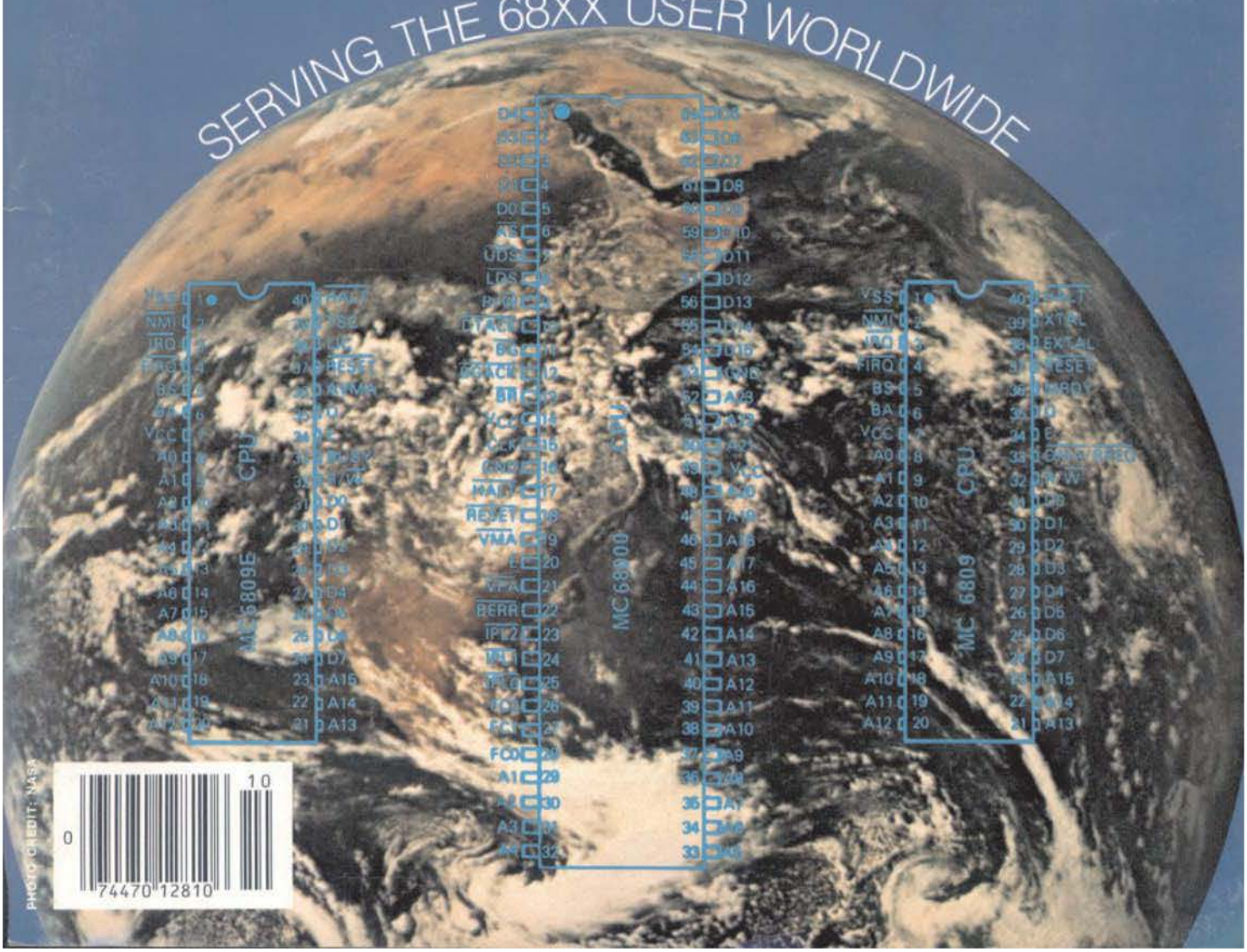


PHOTO CREDIT: NASA

0 74470 12810 10

WE DON'T PLAY GAMES



X-12+ A SERIOUS COMPUTER IN A DESKTOP PACKAGE

Multiprocessor Technology - Combination of 8, 16 and 32 bit types

1.0 Megabyte Memory - Insures no limitation on programs

"Winchester" Disk System - Fast response, large storage capacity

UniFlex* Operating System - The standard of comparison

Hardware Floating Point - Unmatched speed in a small system

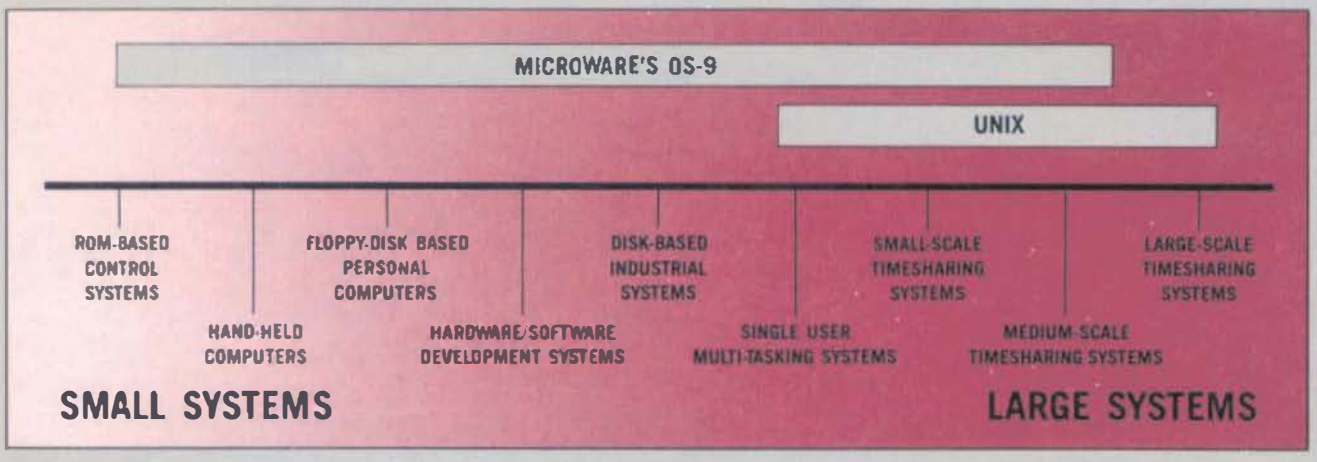
Up to Three Terminals - Instant expansion

* Trademark of Technical Systems Consultants

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Only Microware's OS-9 Operating System Covers the Entire 68000 Spectrum



Is complicated software and expensive hardware keeping you back from Unix? Look into OS-9, the operating system from Microware that gives 68000 systems a Unix-style environment with much less overhead and complexity.

OS-9 is versatile, inexpensive, and delivers outstanding performance on any size system. The OS-9 executive is much smaller and far more efficient than Unix because it's written in fast, compact assembly language, making it ideal for critical real-time applications. OS-9 can run on a broad range of 8 to 32 bit systems based on the 68000 or 6809 family MPUs from ROM-based industrial controllers up to large multiuser systems.

Key OS-9 Features At A Glance

- Compact (16K) ROMable executive written in assembly language
- User "shell" and complete utility set written in C
- C-source code level compatibility with Unix
- Full Multitasking/multiuser capabilities
- Modular design - extremely easy to adapt, modify, or expand
- Unix-type tree structured file system
- Rugged "crash-proof" file structure with record locking
- Works well with floppy disk or ROM-based systems
- Uses hardware or software memory management
- High performance C, Pascal, Basic and Cobol compilers

OS-9'S OUTSTANDING C COMPILER IS YOUR BRIDGE TO UNIX

Microware's C compiler technology is another OS-9 advantage. The compiler produces extremely fast, compact, and ROMable code. You can easily develop and port system or application software back and forth to standard Unix systems. Cross-compiler versions for VAX and PDP-11 make coordinated Unix/OS-9 software development a pleasure.

SUPPORT FOR MODULAR SOFTWARE — AN OS-9 EXCLUSIVE

Comprehensive support for modular software puts OS-9 a generation ahead of other operating systems. It multiplies programmer productivity and memory efficiency. Application software can be built from individually testable software modules including standard "library" modules. The modular structure lets you customize and reconfigure OS-9 for specific hardware easily and quickly.

A SYSTEM WITH A PROVEN TRACK RECORD

Once an underground classic, OS-9 is now a solid hit. Since 1980 OS-9 has been ported to over a hundred 6809 and 68000 systems under license to some of the biggest names in the business. OS-9 has been imbedded in numerous consumer, industrial, and OEM products, and is supported by many independent software suppliers.

microware® OS-9™

MICROWARE SYSTEMS CORPORATION
1866 NW 114th Street
Des Moines, Iowa 50322
Phone 515-224-1929
Telex 910-520-2535

Microware Japan, Ltd
3-8-9 Baraki, Ichikawa City
Chiba 272-01, Japan
Phone 0473(28)4493
Telex 299-3122

OS-9 is a trademark of Microware and Motorola. Unix is a trademark of Bell Labs.

'68' MICRO JOURNAL

Portions of the text for **68 MICRO JOURNAL** was prepared using the following furnished hard/software.

COMPUTERS-HARDWARE
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216
S09-5/8 DMF disk-CDS1-8212W-Sprint 3 Printer

GIMIX Inc.
1337 West 37th Place
Chicago, IL 60609
Super Mainframe-OS9-FLEX-Assorted Hardware

EDITORS-WORD PROCESSORS
Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, NC 27514
FLEX-Editor-Processor

Great Plains Computer Company, Inc.
PO Box 916
Idaho Falls, ID 83401
STYLO-Mail Merge

Editorial Staff

Don Williams Sr.	Publisher
Larry E. Williams	Executive Editor
Tom E. Williams	Production Editor
Robert (Bob) Nay	Color Editor

Administrative Staff

Mary Robertson	Office Manager
Penny Williams	Subscriptions
Michael Westfall	Shipping/Rec.
Christine Kocher	Accounting

Contributing Editors

Ron Anderson
Norm Commo
Peter Dibble
Dr. Theo Elbert
William E. Fisher
Dr. E.M. Pass

Special Technical Projects

Clay Abrams K6AEP
Tom Hunt

CONTENTS

Vol.VI, Issue IX October 84

FLEX USER Notes	8	Anderson
OS9 USER Notes	11	Dibble
C USER Notes	14	Pass
CAT Utility for FLEX 1 2 & 9	17	Puglia & Taylor
LETNUM Rountine	24	Gass
Microware Relocatable Assembler & Linker	25	Pass
68K Experiences	30	Salzwedel
Cyclic Redundancy Check	33	Magnus
Music on the 68000	34	Dildy
KBASIC	41	
Bit Bucket	43	
Corrections	47	
Classifieds	48	

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
Ph (615)842-4600 TELEX 558 414 PVT BTH

Copyrighted 1984 by
Computer Publishing Inc. (CPI)

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES
USA
1-Year $24.50 2-Years $42.50 3-Years $64.50
FOREIGN
See Page 60

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors **full name, address, date and telephone number.** It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands **ONLY** should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continous text.

All articles submitted on diskettes should be in TSC FLEX™ format, either FLEX2 6800, or FLEX9 6809 any version.

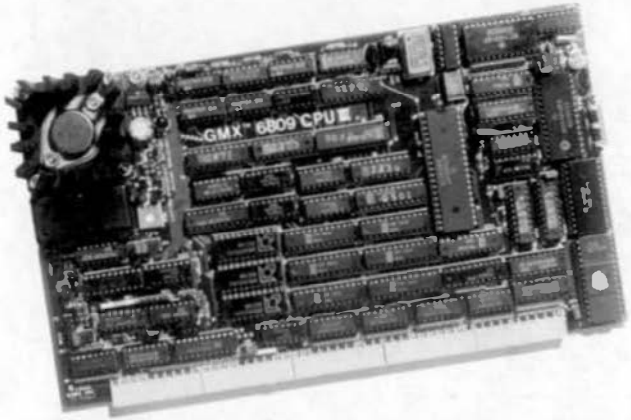
If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). **Please use a dark ribbon!**

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less $7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

GIMIX STATE OF THE ART 6809 SYSTEMS FOR THE SERIOUS USER.



GIMIX has 19MB or high performance 47MB Winchester Drive Systems and/or Floppy Disk Drive Systems.

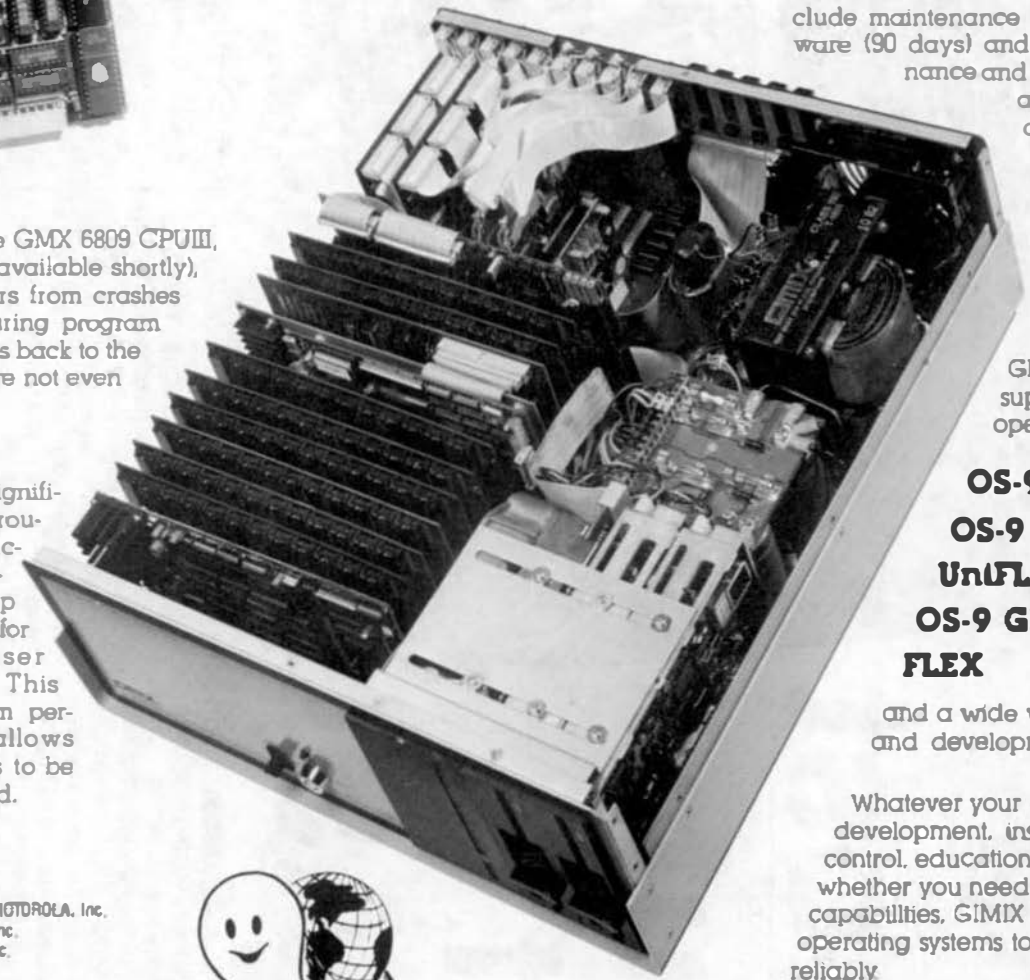
For the user who appreciates the need for a bus structured system using STATIC RAM and powered by a ferro resonant constant voltage transformer.

GIMIX has single user systems that can run both FLEX and OS-9 or Multi user systems for use with UniFLEX or OS-9.

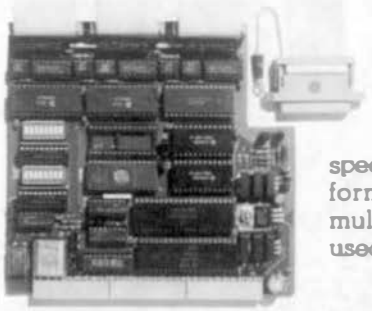
GIMIX versions of OS9 and UniFLEX include maintenance and support by Microware (90 days) and TSC (1 year). Maintenance and support after this period are available at extra cost.

(NOTE: this support and maintenance is only for use with approved GIMIX hardware)

For the ultimate in performance, the Unique GMX 6809 CPUIII, using either OS-9-GMXIII or UniFLEX GMXIII (available shortly), gives protection to the system and other users from crashes caused by defective user programs. e.g. During program development, a programmer who crashes goes back to the shell or the debugger, while the other users are not even aware anything occurred.



The intelligent serial I/O processor boards significantly reduce system overhead by handling routine I/O functions, thereby freeing up the host CPU for running user programs. This speeds up system performance and allows multiple terminals to be used at 19.2K baud.



GIMIX 6809 systems support five predominant operating systems:

OS-9 GMX III,
OS-9 GMX II,
UniFLEX,
OS-9 GMX I,
FLEX

and a wide variety of languages and development software.

Whatever your application: software development, instrumentation, process control, educational, scientific or business; whether you need single or multi-user capabilities, GIMIX has hardware and the operating systems to get the job done reliably.

Please phone or write if you need further information.

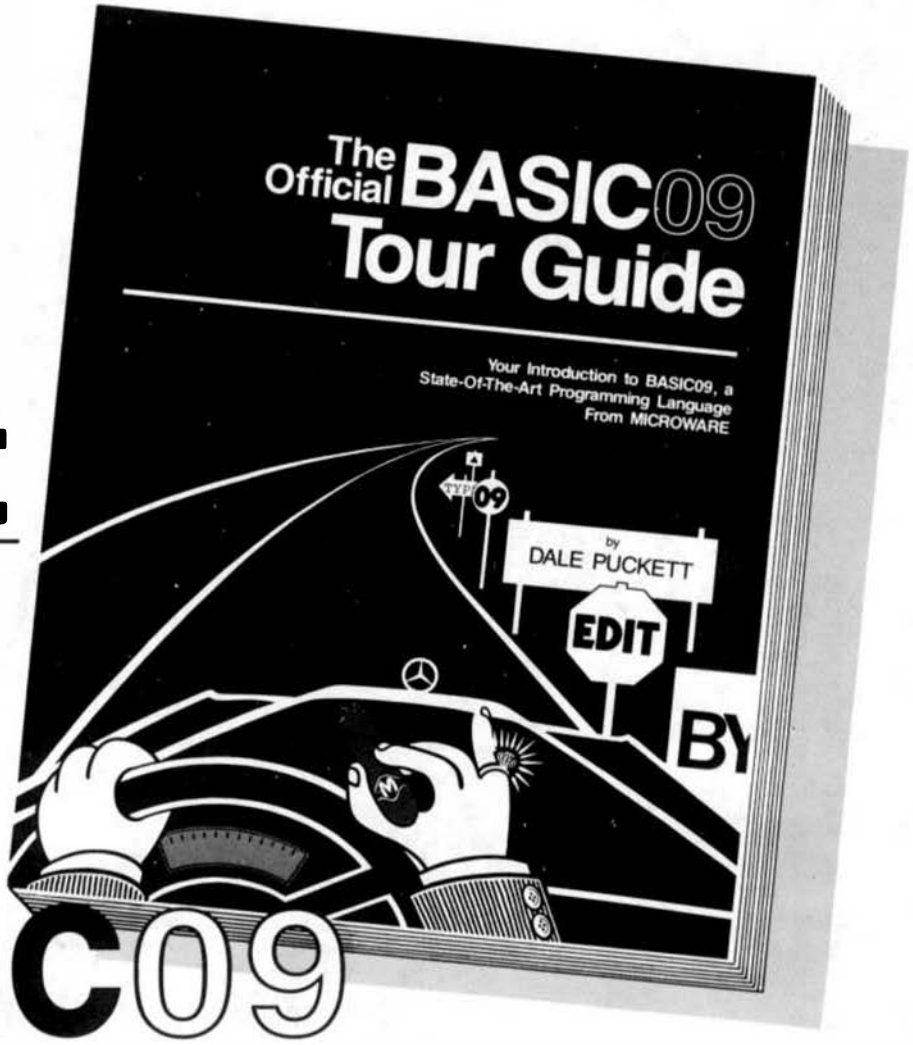
BASIC-09 and OS-9 are trademarks of Microware Systems Corp. and MOTOROLA, Inc.
FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.
GIMIX, GHOST, GMX, CLASSY CHASSIS, are trademarks of GIMIX, Inc.

GIMIX INC.

1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609 • (312) 927-5510 • TWX 910-221-4055

©1983 GIMIX Inc.

Get the most out of BASIC09



The **OFFICIAL BASIC09 TOUR GUIDE** is skillfully written in a friendly and easy-to-read style. Just perfect for those new to computers and to BASIC09. It's also a *valuable reference book* for programmers, engineers, students and hobbyists, providing an in-depth look at BASIC09 plus an overview of the OS-9 operating system. Comprehensive reference sections on BASIC09 and OS-9 commands are also included.

The book "maps" out your route through the Mercedes of Basics . . . BASIC09 and puts you in the driver's seat in no time. Fasten your seatbelt, sit back and enjoy the ride to perfecting your programming skills.

MICROWARE . . .

The **OFFICIAL BASIC09 TOUR GUIDE** comes from the people who wrote BASIC09. As the leader in 6809 system software, we at MICROWARE care about our users and want to help you get the most from our products.

It's Easy to Order.

Phone orders are accepted from MasterCard or VISA cardholders or for COD shipment. You can also order by mail using the coupon below. Quantity discounts are available to educational organizations and dealers. For further information contact Microware.

microware®

Specialists in system software for 68-family microprocessors since 1977.

OS-9 and BASIC09 are trademarks of Microware and Motorola

Microware Systems Corporation
1866 N.W. 114th Street
Des Moines, Iowa 50322
Telephone 515/224-1929
Telex 910-520-2535

Please send _____ copies of the **Basic09 Tour Guide** book at $18.95 each. Add $2.00 for UPS shipping in the U.S. or $5.00 for overseas air mail per book. Iowa residents add 4% sales tax.

Name _____

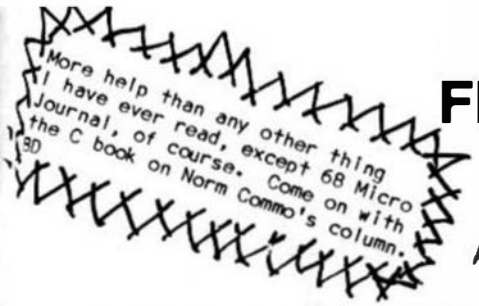
Address _____

City _____

State _____ Zip _____

☐ I have enclosed a check

☐ Charge to my bank card:
MasterCard ☐ VISA ☐
Card Number _____
Expiration _____



FLEX™ USER NOTES
THE 6800-6809 BOOK

By: Ronald W. Anderson
As published in 68 MICRO JOURNAL™



The publishers of 68 MICRO JOURNAL are proud to announce the publication of Ron Anderson's **FLEX USER NOTES,** in book form. This popular monthly column has been a regular feature in 68 MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. Now all his columns are being published, in whole, as the most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

As a SPECIAL BONUS all the source listing in the book will be available on disk for the low price of: FLEX™ format only — 5" $12.95 — 8" $16.95 plus $2.50 shipping and handling, if ordered with the book. If ordered separately the price of the disks will be: 5" $17.95 — 8" $19.95 plus $2.50 shipping and handling.

Listed below are a few of the **TEXT** files included in the book and on diskette.

All TEXT files in the book are on the disks.

LOGO.C1	File load program to offset memory — ASM PIC
MEMOVE.C1	Memory move program — ASM PIC
DUMP.C1	Printer dump program — uses LOGO — ASM PIC
SUBTEST.C1	Simulation of 6800 code to 6809, show differences — ASM
TERMEM.C2	Modem input to disk (or other port input to disk) — ASM
M.C2	Output a file to modem (or another port) — ASM
PRINT.C3	Parallel (enhanced) printer driver — ASM
MODEM.C2	TTL output to CRT and modem (or other port) — ASM
SCIPKG.C1	Scientific math routines — PASCAL
U.C4	Mini-monitor, disk resident, many useful functions — ASM
PRINT.C4	Parallel printer driver, without PFLAG — ASM
SET.C5	Set printer modes — ASM
SETBAS1.C5	Set printer modes — A-BASIC
	(And many more)

**Over 30 TEXT files included in ASM (assembler) — PASCAL — PIC (position independent code) TSC BASIC-C, etc.

NOTE: .C1,.C2, etc. = Chapter 1, Chapter 2, etc.

This will be a limited run and we cannot guarantee that supplies will last long. Order now for early delivery.



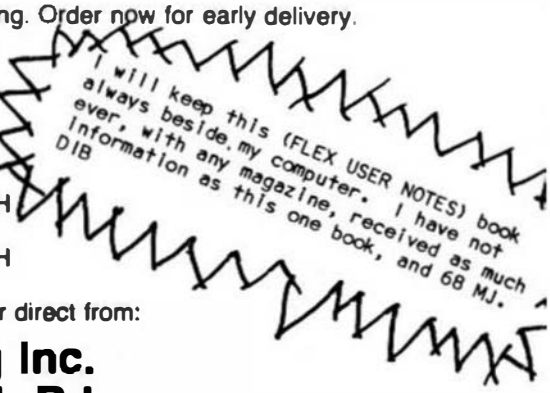
Foreign Orders Add $4.50 S/H

Softcover — Large Format

Book only: **$7.95** + $2.50 S/H

With disk: 5" **$20.90** + $2.50 S/H

With disk: 8" **$22.90** + $2.50 S/H



See your local S50 dealer/bookstore or order direct from:

Computer Publishing Inc.
5900 Cassandra Smith Rd.
Hixson, TN 37343
(615) 842-4601

TELEX 558 414 PVT BTH

™FLEX is a trademark of Technical Systems Consultants

$4,325
FOR A WORLD-CLASS SS-50 COMPUTER

Smoke Signal's VAR/68™ gives you:

- Fabled Chieftain performance that led the pack in tough Benchmark surveys
- Integrated, easy-to-use software that covers your *complete* business needs
- Proven reliability backed by our exclusive Endurance-Certification Program
- Extremely good looks and unsurpassed operator comfort



$4,325: A PRICE CALCULATED TO GET YOU HOOKED ON THIS BLOCKBUSTER SS-50

That price buys you a VAR/68 computer with multi-user, multi-tasking capabilities, and an ergonomically designed terminal. You get 128K RAM—expandable to 1mb. Eight serial ports, up to 16 if desired. Two parallel ports—and more are available. Plus a long list of other impressive capabilities.

Smoke Signal's experience allows us to offer OS-9 and other UNIX-like, and multi-user operating systems.

The styling is completely new—fashioned for the utmost in operator comfort. And it's remarkably compact. VAR/68 is a combination of great performance and good looks demanded by the office of today.

VAR/68 IS TOUGH, BUT SMOKE SIGNAL GIVES YOU *EXTRA* PROTECTION

(1) Your new computer is Endurance-Certified before delivery. That's an exclusive quality-assurance process that guarantees perfect operations from day one.

(2) Our Advance-Replacement program is yours for a low fixed charge. (3) You get instant diagnostic service by telephone. It's free. (4) Normal repairs are handled with super speed. (5) Software *and* hardware support are part of doing business with Smoke Signal.

TOTAL INTEGRATED SOFTWARE GIVES YOUR BUSINESS *SOLUTIONS* INSTEAD OF PROBLEMS

Powerful business application programs are ingeniously interlinked to give even untrained operators a quick, smooth upper hand. The VAR/68 is a joy for first-time users, and an unprecedented productivity tool for *anyone* who wants new dimensions of control over critical business matters.

This screen tells part of the story:



GET A BIG DISCOUNT ON YOUR INITIAL ORDER

Most re-sellers can save up to 42 percent—even on small orders. Smoke Signal's price schedule is a powerful profit-maker for dealers of almost every description.

CALL SMOKE SIGNAL OR WRITE FOR MORE INFORMATION ON THE VAR/68 COMPUTER FAMILY

VAR/68™
BY SMOKE SIGNAL
The computer designed for your specialized application software

SMOKE SIGNAL
Products and Support for VARs

31336 Via Colinas • Westlake Village, CA 91362-3984 • (818) 889-9340

THE 68000 FROM SMOKE SIGNAL!

ADD 68000 AND UNIX™ * TO YOUR EXISTING SS-50 COMPUTER AT PRICES 50% TO 75% OFF LIST

THANK YOU

Seven years ago, Smoke Signal was founded to sell state-of-the-art computer products, by mail, to individual professional programmers and hardware engineers. At that time, most big companies did not believe in the power or future of microcomputers for serious computing applications. Only after you, the individual computer user, proved the viability of the microcomputer was Smoke Signal able to sell systems for business uses. However, as we progressed to become the leader in SS-50 systems, we had to add the sales and technical support services demanded by these business customers — and our prices for complete systems reflected these added costs.

With the introduction of our 68000 products, we wanted to find a way to say thanks to you, our original customers, the individual computer users, and still offer complete sales and technical support to our business customers for complete systems. We think this offer accomplishes both of these goals. We are offering you a choice of upgrade kits that will bring any SS-50 computer up to the electrical equivalent of our complete 68000 computer systems at prices far below complete system prices. In fact, the prices offered are **50%** or more **off** our normally low prices for the components contained in the upgrade kits.

This special offer is limited to one upgrade kit per customer and is our way of saying thanks to those of you who had confidence in us from the beginning.

THE UPGRADES

The following upgrade kits were designed so that any SS-50 system can be upgraded to 68000/UNIX.

SWTP UPGRADE..... $2,800.00
Contains: LMB-1A SS-50C Motherboard, DCB-4A floppy controller, PSA-1 Winchester/Tape DMA interface, SCB-68K 68000 CPU, SER-2 dual serial board, 5Mb Winchester and controller, power supply, all cables, and REGULUS.

GIMIX UPGRADE..... $2,500.00
Contains: Same as SWTP Upgrade except allows you to use your GIMIX motherboard, serial board and Winchester power supply.

Users of standard SMOKE SIGNAL systems may choose one of the following upgrade kits:

For SSB floppy based systems:

SS-FD UPGRADE $2,100.00
Contains: SCB-68K 68008 CPU, PSA-1 Winchester/Tape DMA interface, 5Mb Winchester and controller, power supply, all cables, and REGULUS.

For SSB Winchester based systems:

SS-HD UPGRADE..... $500.00
Contains: SCB-68K 68008 CPU and REGULUS.

COMPLETE SYSTEMS

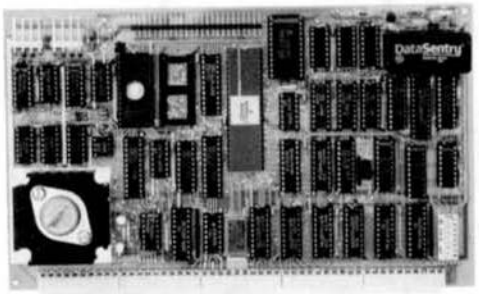
SMOKE SIGNAL is also making available complete VAR/68K™ systems at dramatic discounts. This offer is only available through SMOKE SIGNAL dealers. Contact SMOKE SIGNAL directly for information about how to order a complete VAR/68K system.

RULES OF THE OFFER

1) Limit, one upgrade system per customer.
2) Prices valid through December 31, 1984.
3) Orders must be accompanied by full payment in the form of individual check or credit card authorization.
4) Support will only be provided for systems containing the following SMOKE SIGNAL boards: SCB-68K, DCB-4A, PSA-1, and a motherboard such as the LMB-1A with extended addressing and main terminal I/O at FF7E8.
5) While we feel that most static RAM boards will work with these upgrades, we only guaranty compatibility with systems containing SMOKE SIGNAL static or dynamic RAM.

VAR/68K is a trademark of Smoke Signal.
REGULUS is a registered trademark of Alcyon Corp.; UNIX is a registered trademark of Bell Laboratories; OS9 and OS9/68K are trademarks of Microware; MACSBUG is a trademark of Motorola Inc.

***Regulus the OS offered is UNIX Compatible**



PRODUCTS

The heart of all these upgrade kits is SMOKE SIGNAL'S new SCB-68K 8MHz 68008 CPU Board. This standard (5½" x 9") board will replace a SCB-69 CPU Board in any SMOKE SIGNAL computer with current revision boards. This board contains a real-time clock with battery back-up, 2 EPROM slots for up to 64K bytes of storage, a MACSBUG™ type monitor along with an auto boot loader and a mnemonic disassembler, plus many more features.

All upgrades also come standard with REGULUS™, a UNIX like operating system which is totally compatible with UNIX. REGULUS supports real-time tasks, shared memory, record locking and contains a shell similar to the Berkeley C shell. Along with the operating system, you get C, an editor, assembler, linking loader, interactive debugger and a word processor.

SMOKE SIGNAL is also including in many of the kits the DCB-4A double density floppy controller which can handle up to four 5" and four 8" floppies and contains 1K of buffer RAM for fast disk transfers; the PSA-1 Winchester/Tape DMA interface board which has taps for SASI and Priam disk interfaces as well as a tap for 90 ips tape streamers which are supported under both REGULUS and OS9™; either a M-256-X or M-512-X dynamic RAM board with over two years of field proven reliability; and the LMB-1A heavy duty motherboard with gold plated connectors, extended addressing and on-board baud rate generator with ten selectable baud rates.

SOFTWARE

Software and Software Support is available only from Smoke Signal dealers. Spread Sheet, Word-Processing, Relational Database, C, Basic and Cobol are all available now. Additional system's software is becoming available every day because of the UNIX compatibility.

SMOKE SIGNAL dealers are also offering Microware's OS9/68K™ to purchasers of these upgrade kits. SMOKE SIGNAL will offer other Microware 68000 products as they become available.

SUPPORT

Even at these "lower than PC" prices, we're not going to leave you with "PC" type support. We've arranged with one of our very technically qualified dealers to provide you with add-on software and technical support. In addition to answering your questions on how to convert your system to the 68000, he has a group of his customers who are themselves computer experts who are joining in a network that will help with even the most technical questions. We hope you will contribute your ideas to the network so that we all can benefit from new and fresh thinking. Complete details of the support available are included with the upgrade systems.

ORDER FORM

Fill in your name, address and phone number below. Your order will be shipped UPS so please do not use P.O. Box. Check items being ordered on form. Add prices for all items selected. CA residents must add 6% for sales tax. Total the amount for your order and check payment method below.

Name _____
Address _____
City, State, Zip _____
Phone _____

Payment: ☐ Enclosed Check
☐ VISA
Mastercard
Card # _____
Exp. Date _____
Signature _____

☐ SS-FD UPGRADE	$2100	_____
☐ SS-HD UPGRADE	500	_____
☐ SWTP UPGRADE	2800	_____
☐ GIMIX UPGRADE	2500	_____
☐ M-256-X RAM	648	_____
☐ M-512-X RAM	948	_____
☐ SER-2 I/O	65	_____
☐ 20Mb HARD DISK (instead of 5Mb)	800	_____
Sub Total		_____
CA residents add 6%		_____
Total		_____

SEND COMPLETED ORDER FORM TO:
SMOKE SIGNAL
31336 Via Colinas, Westlake Village, CA 91362-3954
(818) 889-9340

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

An Old Friend

In one sense it seems like a very long time ago, and in another, it might have been a few months ago. I'm speaking of my introduction to the 6800 Microprocessor and microcomputing in general. I owe a great deal to the folks at SWTPc who manufactured the kit that I put together to make my first computer with a real live CRT. I had previously operated a KIM-1 with a hexadecimal key pad and some LED displays, and that was a real education in machine language programming. Soon after seeing the potential uses for microprocessors in my line of engineering, I ordered the SWTPc computer, a CT-1024 terminal, a cassette interface, and a PR-40 (40 column) printer. It was actually in early 1977 that I placed my order. SWTPc at the time, was more or less swamped with orders, and it was late June when the last components of the system arrived.

Meanwhile, as the pieces started to arrive, I built the "terminal" kit (a terminal less the monitor, which I used with a RF modulator and an old black and white TV set. One of the early packages was, believe it or not, the software that I had included in my order. There was an Assembler called CORES, which made no sense to me at the time, until I realized that it should have been CO-RES, indicating that the Assembler and Editor were CO-RESident in memory at the same time. In the days of 300 baud tape loading, that was a nice feature. Also with this package, I received what SWTPc then called "8K BASIC". The manual and the cassette arrived, and I had a month or more to read the manual every evening and dream of the day the hardware would arrive and I could try my hand at some programming. Meanwhile I had gotten the terminal working with the KIM-1, and had a version of Tiny BASIC running, so I could learn some elementary principles of writing programs in BASIC.

The point of my reminiscence however, is the 8K BASIC. The manual said that it was written by Bob Uiterwyk, and I later learned that the same BASIC with very slight modifications, was the one that was sold by Motorola for their "Exorciser" development system. I have to say that this BASIC was responsible for the early reputation the 6800 had of being very slow. While Microsoft had come along with a 6 digit BINARY floating point package in their BASIC, 8K BASIC had a 9 digit BCD (Binary Coded Decimal) math package. You may realize that BCD arithmetic is a lot slower than Binary arithmetic in most computers because computers "think" in Binary. BCD is inefficient, because only ten of every possible 16 codes are used for numbers. The result of operations has to be "decimal adjusted" (the 6800 and 6809 have a DAA instruction to do just that). Negative numbers in 8K BASIC are represented as "9's complement" numbers. It takes more time and effort to calculate the 9's complement than it does in a binary package to complement and add 1, for example.

Notwithstanding the disadvantages of the BCD math, there are some offsetting advangages. First of all, since our money calculations are carried out in decimal, there are no errors introduced by the way the numbers are represented. It is a fact that there is no accurate representation of the value 0.1 in binary. It becomes an endless stream of 1's and zero's in a repeating pattern just as 1/11 becomes 0.09090909090909... in decimal representation. Because we can never include enough binary digits (bits) to represent 0.1 accurately in binary form, converting 0.1 to binary and back to decimal usually results in a value of 0.099999999.. Some of the better binary math package authors recognize this defect and calculate with a "guard digit", rounding the result. This usually takes care of the imperfect representation problem to the limit of the number of digits represented in the package. The extra code and effort somewhat offset the advantages of the fast binary math, and for scientific calculations, the "rounding" or "representational" errors are not (usually) serious or even significant.

The fact is, then, that BCD math packages, because they can represent decimal numbers and fractions exactly, are more accurate and thus desirable for monetary calculations. In either type of math package, doubling the number of digits represented, generally slows the calculations down by a factor of four. That is, the number of operations that must be done increases as about the square of the number of digits carried in the calculations.

With this background, I can continue the story. I learned to like everything about that BASIC but its speed. Later when FLEX came along, disk file handling capebilities were added. It had excellent file handling which included an EOF test. IF EOF(1)=1 THEN CLOSE #1. I wrote lots of programs in that BASIC, and essentially learned how to program using it. Some time later, TSC released their 6 digit BASIC and their Extended BASIC with some 16+ digits of accuracy. Both of the new BASIC interpreters used binary math, and both were many times faster than the old 8K version, and for that reason and the greater accuracy of Extended BASIC, it was adopted immediately by the business software suppliers as the language in which they wrote their software.

I remember being particularly upset at TSC for not having included an EOF test. Any loop that was reading a sequential file had to be preceded by an ON ERROR statement, that pointed to a line where the code would be something like IF ERR=8 THEN CLOSE #1 : RESUME 100. 8K BASIC handled end of file much more cleanly by allowing inclusion within the file read loop of IF EOF(1)=1 THEN CLOSE #1: GOTO 100.

What seemed even worse at the time was the awkwardness of writing a number of variables to a data file. Both interpreters use ASCII data files, but old 8K had a "WRITE" keyword. WRITE #1, A,B,C resulted in the same thing as the much more difficult PRINT #1,A;",";B;",";C. The values of the three variables A, B, and C were written to the file, separated by commas, with a CR at the end. 8K BASIC used the special WRITE statement to tell it to separate the variables with commas. Extended BASIC uses the standard PRINT statement, and you have to tell it literally just what you want "PRINTed" to the file.

8K BASIC has the standard features of BASIC, including its 9 digit arithmetic. The Scientific functions standard to BASIC are all present, and all are accurate to 7 digits. Though 8K doesn't have PRINT USING implemented, it does have a way to set the digits printed after the decimal point. DIGITS=3 will accomplish that. You can also set the length of strings and the maximum line length for output. You output to a printer or other device by switching ports with a PORT=7 for a parallel printer, or perhaps a PORT=0 for a modem or serial printer.

Why am I going into all this? It seems that 8K BASIC is not dead. Southeast Media has resurrected it, added a few new features, optimized it for 6809 rather than just reassembling it with a 6809 assembler. The new features include AUTO, which generates line numbers starting at 10 and incrementing by tens as you type in your program. They have a RENUM utility that works like the one that comes with Extended BASIC, except that it always starts with line 10 and increments by tens. The original had a few obscure bugs in some of the scientific functions, which have been cleaned up. In general, the package is complete and very nice to use.

If you want a pleasant dialect of BASIC with medium precision you might be interested in this one. As I indicated above, it is great with sequential files. A caution is in order, however. Random files are NOT supported. This BASIC was written long before FLEX came along with Random File capability. Remember too, that you must expect fairly slow execution. Scientific functions take a significant fraction of a second to calculate. One other omission is the ELSE feature. The syntax includes only IF THEN, but the object of the statement can be a statement itself, unlike some of the very early BASICs in which the object of THEN could only be a line number. This is more of an inconvenience than a lack. You can duplicate the action of IF THEN ELSE easily, but the result is not so clearly an IF THEN ELSE function as when the keywords are available. IF THEN doesn't have the AND and OR capability for complex conditions either, but the AND can be simulated with "run on" IF THEN, as in:

```
100 IF A=B THEN IF C=D THEN ...
```

You can use this syntax for any number of AND conditions. You can OR by using multiple statements as in:

```
100 IF A=B THEN GOSUB 1000
110 IF C=D THEN GOSUB 1000
```

Remember also, on the positive side, that this package is slightly over 8K in size compared to Extended BASIC's 22K. There is a lot more room for program when you use this one. If this extra space allows you to avoid "CHAINing" programs, the speed might turn out to be less a disadvantage than you might think. Of course if you need the speed or 16 digit accuracy, Extended BASIC is the only reasonable choice of BASIC interpreters for you. This BASIC provides us with still another choice of tools, so that we can choose the tool best fitted for the job.

Vacation Experiences

I've just returned from a three week vacation to be reminded with a jolt that I had not finished this column. Fortunately I stay a month ahead (usually) so I will be able to finish this one and work on getting ahead again in the next few weeks.

After five years of corresponding with my "printer pal" Art Weller, we managed to get together. Our family stayed at the Weller's house for three days and we had a vacation together, visiting Carlsbad Caverns, touring Juarez, and breezing around El Paso. It was a most enjoyable stay and now I can picture Art and Jane Weller and their two poodles when I receive letters from them. Somehow we managed to limit our sitting in front of a computer to one short afternoon while the ladies were out shopping. I guess we cover the computer subjects by letter, and didn't have a lot of catching up to do in that area.

I have a number of new products in the area of software to review, and one hardware item. Due to my seeming lack of software subject matter this time, and the urgency of getting this out quickly, I am going to incorporate a review at this point.

Windrush Universal Eprom Programmer II

This is a review of the new Windrush Universal EPROM programmer. This unit is supplied as a new and upgraded model of their "All in Two" EPROM programmer. It arrived packed VERY well in bubble pack and plastic "peanuts". I was impressed, as always, by the considerable care Windrush has taken to provide complete and well organized documentation. The programmer is sold in a number of different configurations for compatibility with the SS-30 SS-50 bus, the Motorola Bus and the "Eurocard" bus. I received the SS-30 interface version.

The programmer itself is in two parts. The first is the interface card that plugs into the SS-30 bus. The card contains only a MC6821 "PIA" and the connector for the braided ribbon cable that leads to the working end of the system, called the "programming pod". The pod is enclosed in a nice injection molded plastic case that looks suspiciously like the case that Radio Shack supplies for their disk interface for the color computer. The case has a 28 pin zero insertion force socket, a pair of leds that tell you what is going on with regard to Vcc and programming voltages, and a set of DIP switches that take the place of numerous "personality modules" of the sort found on some other EPROM programmers.

The software is not quite as smart as some that I have been using (written locally) but I'm probably spoiled by that software. so I will forget it and talk about what Windrush has supplied first, then make a few suggestions.

The Manual is superb if I haven't already indicated as much. One reading of the section of the manual regarding operation, and I was off. The user simply types UPRII, and the system is operative. The first menu is a list of PROM types from which to choose. The programmer and its software now support the intelligent programming algorithm, which tests each location for being programmed, and then "overprograms" that address by a fixed percentage of the time taken to program it. I didn't have any EPROMS available that will program with the fast algorithm, so I couldn't immediately test that feature. At any rate, once the PROM type is selected, you may load the program to be "burned" into the PROM buffer with a LOAD utility. The software provides features such as the ability to read a PROM to the buffer, copy a PROM, dump the PROM buffer, fill the PROM buffer with a specified byte, (usually $FF so that the unused locations are not programmed, since $FF is the erased condition).

The user may examine and change the hex values in the PROM buffer, and he may test the PROM for being erased, though the "program" function does that before allowing the user to program a PROM. One nice feature of the software is that after the PROM type is specified, the software presents a picture of the proper positions of the DIP switches on the pod. The user can set and verify them and then proceed to the next menu described above.

The case has a 28 pin zero insertion force socket, a pair of leds that tell you what is going on with regard to Vcc and programming voltages, and a set of DIP switches that take the place of numerous "personality modules" of the sort found on some other EPROM programmers.

The software is not quite as smart as some that I have been using (written locally) but I'm probably spoiled by that software, so I will forget it and talk about what Windrush has supplied first, then make a few suggestions.

The Manual is superb if I haven't already indicated as much. One reading of the section of the manual regarding operation, and I was off. The user simply types UPRII, and the system is operative. The first menu is a list of PROM types from which to choose. The programmer and its software now support the intelligent programming algorithm, which tests each location for being programmed, and then "overprograms" that address by a fixed percentage of the time taken to program it. I didn't have any EPROMS available that will program with the fast algorithm, so I couldn't immediately test that feature. At any rate, once the PROM type is selected, you may load the program to be "burned" into the PROM buffer with a LOAD utility. The software provides features such as the ability to read a PROM to the buffer, copy a PROM, dump the PROM buffer, fill the PROM buffer with a specified byte, (usually $FF so that the unused locations are not programmed, since $FF is the erased condition).

The user may examine and change the hex values in the PROM buffer, and he may test the PROM for being erased, though the "program" function does that before allowing the user to program a PROM. One nice feature of the software is that after the PROM type is specified, the software presents a picture of the proper positions of the DIP switches on the pod. The user can set and verify them and then proceed to the next menu described above.

I have 2732 PROMs available, and I was able to program one on the first try. The programmer supports the following PROM types:
2758, 2716, 2732, 2532, 2732A, 68764, 2764 (TI), 2764 (Intel), 2764A, 27128, 27128A, and 27256. It is stressed in the manual that the A versions use a different (usually lower) programming voltage, and programming them with a non-A version setup, will cause their destruction. A few compatible types are supported but not shown in the menu. These include the 2508, 2516, and MC68766. The programmer is NOT compatible with any of the earlier TRI-VOLT EPROMS, and a caution is included in the manual, that trying to program them may damage the programmer.

The manual also cautions that the use of the intelligent programming algorithm on PROMs not designed for it, might (and probably will) damage those PROMs. The reason is that VCC is raised to 6 volts while using the intelligent algorithm so that a higher threshold for '0' is provided while programming and testing. INTEL's PROMS are designed for such programming, and others may also be so designed but it is wise to consult the data sheet for the particular device. Only the devices of 2764 size or larger support this programming method. The manual indicates that a 2764 in the intelligent programming mode will program about twice as fast as a 2732 in the normal mode. The manual also indicates that the unit is more or less ready to accept the newest 27512 PROMs when they are available. Some SIMPLE modifications will need to be done to the pod circuitry due to an incompatible pin-out.

One thing I found a little peculiar was the default address of the PIA that controls the programmer. There are actually two command files provided, that have the PIA at address $E070 and $E090 respectively. The $E090 address is for some Eurocard systems that Windrush supplies, and that address is out of the range of any of the standard hardware in the U.S. (Gimix, Smoke Signal, or SWTPc). The $E070 choice puts the PIA in "Port 7" on systems with 16 addresses per port, and out of the range of I/O for systems with four addresses per port. Port 7 is usually used for a parallel printer interface (true in all the SS-50 bus systems with which I work. The point is minor, since the software looks for a port at the default address and asks for the actual address if one is not found there. Also there is a binary patch file that allows you to specify the actual address of the port in your system, and also to specify some terminal control strings for clearing the screen, homing the cursor, and turning reverse field or half intensity on and off. If you specify these, the screen will be more readable, but the software operates without them by means of scrolling the menu off the screen when another one is written to it.tt

Now for the inevitable "why didn't you?". When you load a file to the buffer, you use a simple "offset LOAD" utility provided, so that you can load your file starting at address $0000, the normal start of the buffer. Suppose you are using 2732 PROMs, which store $1000 bytes each, and the file is longer than $1000 bytes. You burn the first PROM, and then it is necessary to move the data in the buffer (via a a MOVE command built into the programming software) down by $1000 so that you can burn the next PROM. When you specify the PROM type, the software knows how much fits in one PROM, and it in fact tells you how much, right on the screen. It should not be necessary to move the data in the buffer, but only to move a pointer to the start of the data that would occupy the second EPROM for a file larger than $1000. (The default buffer extends from $0000 to $7FFF, and so can hold 32K of "data"). There should be a "NEXT" command that would move the pointer by one "promful" and probably a "first" command to restore the pointer to $0000. In this way, several sets of PROMs could be burned without reloading the file. It would also be nice if the buffer were automatically preloaded with $FF bytes so that the "FILL BUFFER" command could be skipped unless for some reason an odd value is required in unused space. If this were done, the software could also determine how many PROMs would be required for a given program, by the extent of the non-$FF area in the buffer. The manual is singularly lacking in discussion of how to program multiple PROMs from the data in the buffer, but I assume it is presently necessary to move the data in the buffer. End of minor gripes.

I should mention that the DIP switch setup mode is not much (if any), less convenient than unplugging a personality module and plugging in another one. We normally use 2716 and 2732 PROMS exclusively. The difference in the dip switch settings for these two, is the position of two switches. Setting these two switches is less time consuming than changing a personality module. Most of us will not be programming 6 or 8 different PROM types at any one time. Surely we will go toward the larger PROMs over a period of time, but as we add larger ones, we will certainly drop the smaller ones, so that at any one time, we will be programming perhaps three or four types. I should also comment that the operation of the programmer is entirely automatic. Once it is set up, the only operator action is to insert and remove PROMs and answer the prompts of the software to program them. No "program - read" mode switch is present on the pod. All voltages are applied and removed under program control. Two LED indicators do show the status of Vcc and Vpp at any time, and the user is cautioned not to insert or remove a PROM when either LED indicates that a voltage is present. The programmed PROM is verified against the buffer, and any error is reported.

Overall, this is a compact, nicely built, well thought out, and well documented combination of software and hardware.

OS9 USER NOTES

by Peter Dibble
("OS-9 Users Notes" Columnist;
'68' Micro Journal)

Reading Directories

I'm afraid this month's column will be a little short. I just bought a house. Nothing major wrong with it, but I'm living in the first floor while I fix up the upstairs. Piles of boxes are everywhere, and it seems like everything I need is in a box at the center of an unknown pile. This disorder has not helped me get a lot of computing done.

I don't mean to turn this column into a diary, but there are a few other important items. A kitten is helping me write this. I got him to help make my house seem home-like, but he likes to help type. I enjoy his help, but I hope he will switch to sleep-in-the-lap mode soon.

This fall I will finally become a full-time graduate student. I have been studying Computer Science part time for years, but it seemed that the field was moving ahead faster than I was learning it. It is a scary business going back to college after being a working man for years, but I'm fairly quivering with eagerness. I have one more column to write as a free man, then I will be a student. I think I can get permission to keep writing this column. I hope my studies add some spice to my writing.

A fair amount of the OS-9 mail I get asks about special versions of OS-9. Many people have old SWTPc systems they would like to run OS-9 on. There are also a few people with home-brew 6809 systems who'd like to port OS-9. The news for these people is mostly bad.

There used to be a SWTPc version of OS-9 Level One, but I don't think it is sold any more. OS-9 Level Two is sold only through hardware manufacturers, and SWTPc hasn't licensed it. If you have your own home-brew design, you can license OS-9 from Microware, but the price is ridiculous unless you mean to sell it.

Officially there is no way to get OS-9 for your SWTPc, or home-brew machine. Unofficially, there are ways. One of the most important features of OS-9 is its hardware independence. The clock and I/O devices are handled by drivers. The interfaces to the drivers are general enough that any reasonable hardware can be accommodated. Microware will sell the source to several device drivers and a few clock drivers. With a copy of OS-9 for any machine, a working OS-9 to build the new OS-9 on, and a collection of source from Microware it should be possible for an experienced programmer to adapt OS-9 to any 6809-based machine I have heard of.

It isn't hard to buy a copy of OS-9 to customize. Try a few manufactures. When I was building crazy systems I did a lot of business with AAA Chicago Computing; they might be able to help you. You don't care what version of OS-9 you get unless you can get one that is already partly compatible with your system. You'll have to write a clock driver, a disk driver, and, if you use an unusual serial chip, a SCF driver. If you want to adapt Level Two, you'll have to buy a version of OS-9 that is designed for the memory management hardware you have. Memory management is done in the OS-9 kernel (OS9P1). It isn't easy to adapt without lots of source code ... the kind of source Microware sells as part of an OEM license ... very expensive.

If anyone has OS-9 running on unsupported hardware let me know. Microware doesn't officially want to support you, but they might not object if we set up a function of the Users Group to help you out. If there is enough interest, maybe we can find a reliable source of adaptable OS-9. In any case, I'll report any tips you send me in this column.

Directories as Files

A directory is a special kind of file, if they are handled correctly, they can be opened and used without much trouble. If you try to list or dump a directory file, you will have trouble. Directory files can only be opened using the directory access mode; and Dump, List, Copy, and most other OS-9 utilities don't use this mode.

The easiest thing to do with a directory is to simply read it and copy it to standard output. The program called DList copies the current directory to standard output. You can see the contents of the current data directory by assembling DList and typing

OS9: DList ! Dump

Directories contain lots of unprintable characters, so if you don't use Dump to format the output you will get giberish on the screen. You may even make your terminal do strange things.

I have a directory with only the file containing DList in it. I ran DList with the command line:

OS9:DList ! Dump >tmp

The contents of Tmp were

```
Addr  0 1  2 3  4 5  6 7  8 9  A B  C D  E F
0 2 4 6 8 A C E
----  ---- ---- ---- ---- ---- ---- ---- ----
---------------
0000  2EAE 0000 0000 0000 0000 0000 0000 0000
0010  0000 0000 0000 0000 0000 0000 0000 01B9
0020  AE00 0000 0000 0000 0000 0000 0000 0000
0030  0000 0000 0000 0000 0000 0000 0000 054D
0040  746D F000 0000 0000 0000 0000 0000 0000
tmp.............
0050  0000 0000 0000 0000 0000 0000 0000 0765
0060  444C 6973 F443 4830 B700 0000 0000 0000
DListCH07.......
0070  0000 0000 0000 0000 0000 0000 0000 0763
```

I wrapped each line of the tmp onto two lines so it would fit in a column. Each entry in the directory takes two lines in the dump (four lines in this listing.) The first two entries are self-referencing. The .. entry is first; the name ".." (2EAE) is at the beginning of the entry. The disk address of the file descriptor (0001B9) for the parent of this directory is at the end of the entry. The second entry is for "." (AE) which is the alias for this directory. The address associated with that (000540) points to the file descriptor for this directory.

The entry for tmp is for the file I put the dump into. The last entry looks like it is for DListCH07, but if you look at the hex part of the dump you will see that the high bit of the "t" is on, meaning that it is the last character in the string. The characters "CH07" are an artifact of a

previous use of the entry for the file SCRATCH07.

DList could be modified to edit the contents of the directory before passing it to standard output. The first two entries are always for "." and "..". There is usually no need to notice them. There can also be null entries in the directory. When a file is deleted the first byte in the directory entry is set to $00 making it into a null entry. DList could check for null entries and suppress them.

DList2 is an enhanced version of DList. It uses I$Seek to skip the first two entries, then copies all entries that don't start with $00 to standard output.

The next feature to add would be formatting the output so it could be read without using Dump. The address of the file descriptor isn't likely to be worth seeing very often, so the final program, ld, just prints the file names. A useful directory list program needs to be able to list the contents of directories other than the current default so I added that function. Ld can take a directory name on the command line. It wouldn't be too hard to add the x option by opening the directory with the execution attribute, but that's a function I didn't add. The program determines whether a directory name was given by checking the length of the parameter area. If the parameter area is only one byte long, it only contains a carriage return, otherwise it contains a directory name terminated with a carriage return.

It would be nice to add the "e" option to the ld command, but an extended directory involves lots of numbers and dates. The code to format all that information would make a long program. Instead, I have written a Basic09 program that takes the output of DList2 and generates a more extensive report. There is room for lots of improvement in DFormat; I only print the file name, creation date, last modified date, and file size, and I don't sort the list in any special order. Improvements like these are, as they say, "left for the reader."

Each directory entry contains the disk address of the file descriptor sector for the file. The file descriptor contains all the interesting information about a file. We need to read the file descriptor, but all we know is its disk address, and the only way to get at a particular sector on a disk is with physical-sector I/O. Normally physical-sector I/O is done by opening a device; e.g. dump /D0@. Since there is no easy way to find out the name of the drive the directory is on, the /D0@ type of trick isn't useful. .There is an interesting variation on physical-sector I/O which I haven't been able to find documented anywhere. If you open the file @, it will open the drive the data directory is on for physical I/O. If you open it for execution, it will open the drive with the execution directory on it for physical I/O.

Since DList2 is feeding this program, and DList2 can only read the current data directory, DFormat assumes the directory is on the same disk as the data directory.

I used a useful trick from the UNIX ls command. Directory files are indicated by a "/" after them in the listing from DFormat.

If you have a UNIX-like sort program available the combination of DList2 and DFormat can be made even more useful. By sorting on the various fields in the output from DFormat you can get the listing alphabetically by name, by increasing size, or in chronological order.

If you have RunB type DFormat in, save it, and pack it. Then use it with a command line like:
OS9:DList2 1 DFormat If you don't have RunB, you'll need to use something like:
OS9:DList2 1 basic09 DFormat

```
00001                           nam   DList
00002                           ttl   List the Current Directory
00003                           IFP1
00005                           ENDC
00006  0011             type     set   PRGRM+OBJCT
00007  0081             Revs     set   REENT+1
00008  0000 87C00043            mod   MEnd,Name,Type,Revs,Entry,Memsize
00009
00010 0 0000             DPath    rmb   1          Directory path number
00011 0 0001             Buffer   rmb   32         buffer for directory entries
00012 0 0021             Stack    rmb   200
00013 0 00E9             Memsize  equ   .
00014
00015  0000 444C6973     Name     fcs   /DList/
00016  0004 01           Version  fcb   1
00017  0005 2EA0         Dirname  fcs   /. /
00018
00019  0007             Entry
00020  0007 8681                 lda   #DIR.+READ.
00021  0009 308DFFF8             leax  Dirname,PCR
00022  000B 103F84               OS9   I$Open
00023  000E 2510                 bcs   Error
00024  0010 9700                 sta   DPath
00025  0012             RLoop
00026  0012 3041                 leax  Buffer,U
00027  0014 108E0020             ldy   #32
00028  0018 103F89               OS9   I$Read
00029  001B 2508                 bcs   TEof
00030  001D 8601                 lda   #1         Std output
00031  001F 103F8A               OS9   I$Write
00032  0022 2509                 bcs   Error
00033  0024 9600                 lda   DPath
00034  0026 20EA                 bra   RLoop
00035  0028             TEof
00036  0028 C1D3                 cmpb  #E$EOF
00037  002A 2601                 bne   Error
00038  002C 5F                   clrb
00039  002D             Error
00040  002D 103F06               OS9   F$Exit     return
00041  0030 b0C250               EMOD
00042  0043             MEnd     equ   *

00000 error(s)
00000 warning(s)
$0043 00067 program bytes generated
$00E9 00233 data bytes allocated
$223F 08787 bytes used for symbols


00001                           nam   DList2
00002                           ttl   List the Current Directory
00003                           IFP1
00005                           ENDC
00006  0011             type     set   PRGRM+OBJCT
00007  0081             Revs     set   REENT+1
00008  0000 87C0005D            mod   MEnd,Name,Type,Revs,Entry,Memsize
00009
00010 0 0000             DPath    rmb   1          Directory path number
00011 0 0001             Buffer   rmb   32         buffer for directory entries
00012 0 0021             Stack    rmb   200
00013 0 00E9             Memsize  equ   .
00014
00015  0000 444C6973     Name     fcs   /DList2/
00016  0005 01           Version  fcb   1
00017  0006 2EA0         Dirname  fcs   /. /
00018
00019  0008             Entry
00020  0008 8681                 lda   #DIR.+READ. file access code
00021  000A 308DFF98             leax  Dirname,PCR file name "."
00022  000E 103F84               OS9   I$Open
00023  0011 2530                 bcs   Error
00024  0013 9700                 sta   DPath      save the path number
00025  0015 34400                pshs  U          save U
00026  0017 CE0040               ldu   #32*2
```

```
00027   0028 8E0000            ldx   #0
00028   002B 103F88            OS9   I$Seek     skip over . and .. entries
00029   002E 3540              puls  U           restore U
00030   0030 251F              bcs   Error
00031   0032                   RLoop
00032   0032 3041              leax  Buffer,U
00033   0034                   RLoop2
00034   0034 108E0020          ldy   #32
00035   0038 103F89            OS9   I$Read
00036   003B 2506              bcs   TEof
00037   003D 6D41              tst   Buffer,U    null entry?
00038   003F 27F3              beq   RLoop2      yes: skip it and read again
00039   0041 8601              lda   #1          Std output
00040   0043 103F8A            OS9   I$Write
00041   0046 2509              bcs   Error
00042   0048 9600              lda   DPath       directory path
00043   004A 20E6              bra   RLoop       read again
00044   004C                   TEof
00045   004C C1D3              cmpb  #E$EOF      Is this EOF?
00046   004E 2601              bne   Error       no; error
00047   0050 5F                clrb              yes; return happy
00048   0051                   Error
00049   0051 103F06            OS9   F$Exit     return
00050   0054 C80070            EMOD
00051   0057                   MEnd  equ   *

00000 error(s)
00000 warning(s)
$0057 00087 program bytes generated
$00E9 00233 data bytes allocated
$2DE5 08792 bytes used for symbols
```

```
00001                          nam   Id
00002                          ttl   List Files in a Directory
00003                          IFP1
00005                          EMOD
00006   0011           type    set   PRGRM+OBJCT
00007   0081           Revs    set   REENT+1
00008   0000 87C90072          mod   MEnd,Name,Type,Revs,Entry,Memsize
00009
00010 D 0000           DPath   rmb   1         Directory path number
00011 D 0001           Buffer  rmb   32        buffer for directory entries
00012 D 0021           Stack   rmb   200
00013 D 00E9           Memsize equ   .
00014
00015   0002 6CE4      Name    fcs   /Id/
00016   0005 01        Version fcb   1
00017   0006 2EA0      Dirnam  fcs   /. /
00018
00019   0012           Entry
00020   0012 10830001          cmpd  #1        length of parameter string
00021   0016 2204              bhi   DirMdlvn
00022         *          If more than one byte of parameters
00023         *          assume file name on command line.  Otherwise use '.'
00024   0018 308D0FF4          leax  Dirnam,PCR use '.' as directory
00025   001C                   DirMdlvn
00026   001C 8681              lda   #DIR.+READ. file access mode
00027   001E 103F84            OS9   I$Open
00028   0021 2532              bcs   Error
00029   0023 9700              sta   DPath       save the path number
00030   0025 3440              pshs  U           save U
00031   0027 CE0040            ldu   #32*2
00032   002A 8E0000            ldx   #0
00033   002D 103F88            OS9   I$Seek      skip over . and .. entries
00034   0030 3540              puls  U           restore U
00035   0032 2521              bcs   Error
00036   0034                   RLoop
00037   0034 3041              leax  Buffer,U
00038   0036                   RLoop2
00039   0036 108E0020          ldy   #32
00040   003A 103F89            OS9   I$Read
00041   003D 2511              bcs   TEof
00042   003F 6D41              tst   Buffer,U    null entry?
00043   0041 27F3              beq   RLoop2      yes: skip it and read again
00044   0043 8D13              bsr   Edit        Prepare file name for printing
00045   0045 8601              lda   #1          Std output
00046   0047 103F8C            OS9   I$WritLn
00047   004A 2509              bcs   Error
00048   004C 9600              lda   DPath       directory path
00049   004E 20E4              bra   RLoop       read again
```

```
00050   0050           TEof
00051   0050 C1D3              cmpb  #E$EOF      Is this EOF?
00052   0052 2601              bne   Error       no; error
00053   0054 5F                clrb              yes; return happy
00054   0055           Error
00055   0055 103F06            OS9   F$Exit      return
00056   0058           Edit
00057   0058 5F                clrb
00058   0059           ELoop
00059   0059 6D85              tst   B,X
00060   005B 2707              beq   ELoop1
00061   005D 2708              beq   EError      A name can't end in a null
```



```
00062   005F 5C                incb
00063   0060 C11D              cmpb  #29
00064   0062 25F5              blo   ELoop       A name can't be more than 29 b
00065   0064           ELoop1
00066   0064 8600              lda   #$00        <CR>
00067   0066 5C                incb
00068   0067 A785              sta   B,X
00069   0069 39                rts
00070   006A           EError
00071   006A 8600              lda   #$00
00072   006C A784              sta   ,X          Return null line for errors
00073   006E 39                rts
00074   006F 6FA08C            EMOD
00075   0072           MEnd    equ   *

00000 error(s)
00000 warning(s)
$0072 00114 program bytes generated
$00E9 00233 data bytes allocated
$2299 00857 bytes used for symbols
```

```
PROCEDURE DFormat
 0000    TYPE DirEnt=name:STRING[29]; lsn(3):BYTE
 0018    TYPE SegList=lsn(3):BYTE; SegLen:INTEGER
 0031    TYPE fdinf=attr:BYTE; owner:INTEGER; ModDate(5),Link(1..6):BYTE; SegList:ARRAY [1..49] OF SegList
 0070    DIM DirEnt:DirEnt
 0079    DIM FDinf:fdinf
 008C    DIM Real_LSN:REAL
 0089    DIM i,errnum:INTEGER
 0094    DIM PPath:BYTE \DECR physical I/O path number
 00A6    OPEN #PPath,"@":READ
 00B2    LOOP
 00C4      GET 00,DirEnt
 00C9      ON ERROR GOTO 10
 00D3      REM Change name from assembler string format to
 0107      REM Assembler string format by placing a $00 at the end of it.
 013E      FOR i=1 TO 29
 0145        IF(i<=LEN(DirEnt.name)),(i>)127 THEN
 0164          DirEnt.name(i)=0:DirEnt.name(i)
 0177        END IF
 0178        NEXT i
 0186        REM Change the LSN of the FD sector from three bytes
 01B9        REM to a real number
 01CC        Real_LSN=DirEnt.lsn(3)+256*(DirEnt.lsn(2)+256*DirEnt.lsn(1))
 01F4        SEEK #PPath,Real_LSN*256
 0203        GET #PPath,FD
 0210        PRINT DirEnt.name;
 0216        IF FD.attr>127 THEN
 0223          PRINT "/";
 0228        END IF
 0229        PRINT " "; FD.ModDate(0); "/"; FD.ModDate(1); "/"; FD.ModDate(2); " "; FD.ModDate(3); ":"; FD.ModDate(4);
 0265        PRINT "-"; FD.ModDate(5); " "; FD.CDate(0); "/"; FD.CDate(1); "/"; FD.CDate(2); " "; FD.CDate(3);
 02AC        PRINT " "; FD.fd1e0d:d16>*256+lF .File0a:ec1t*256+r8.fd1e0d16
 0302      END LOOP
 030A 10   REM error handler
 0319      errnum=ERR
 0327      IF errnum=211 THEN :REM end of file
 0349        CLOSE #PPath
 0360        END
 0371      ELSE
 0375        PRINT "Error number "; errnum
 0394        END
 039C      END IF
 03AE    END
```

SUPPORT YOUR ADVERTISERS

"C" User Notes

Edgar M. (Bud) Pass, Ph.D.
1454 Latta Lane
Conyers, GA 30207

INTRODUCTION

This month's column discusses problems of execution, transportability, and compatibility among the C compilers available on the 6809. Hopefully, discussing and demonstrating these problems will prompt the developers of these compilers to correct the deficiencies, and the criticisms will be considered constructive.

When appropriate, the latest AT&T UNIX C compiler is used as a basis for comparison. Only the Full C compilers are covered in this discussion, since the Small C compilers all have known limitations and peculiarities, but the Full C compilers claim compatibility with UNIX C and K & R C standards.

All of the problems reported here have been reported to the developers, by myself or by readers who have reported the problems to me.

Some of the problems may have been corrected in the latest versions of the compilers by the time this column appears in print. However, the compilers in the field will not have been corrected, so they should be checked for problems similar to those described below.

The two Full C compiler families discussed are called "McCosh" (marketed by Windrush, MPD, SWTPC, Microware, and Tandy and S.E. MEDIA, CPI*), and "Introl" (marketed by Introl and S.E. MEDIA, CPI*).

BUGS

One of the worst classes of problems is those caused by compiler bugs. The most blatant ones kill the compiler, program, or system.

An example of a compiler bug which may be isolated, demonstrated, and evaluated is as follows:

```
main()
{
        static char s[43] = "S";
}
```

Windrush C v2.5.4 produces the following error message and aborts:

```
Bad code in intermediate file: 5c
```

Introl C and UNIX™ C produces no diagnostics and terminates normally, after a successful compile.

The following example shows a bug which does not terminate the compiler, but may cause programs to malfunction, which may very well present a more serious problem to locate than the previous situation:

```
#include "stdio.h"

struct s1
{
        char c1[8];
        int  i1;
        char h1;
} st1[] =
{
        "item11",1,2,
        "item12",3,4,
        "item13",5,6
};

struct s2
{
        char c2;
        int  i2;
        char h2;
} st2[] =
{
        "item21",1,2,
        "item22",3,4,
        "item23",5,6
};

main()
{
        int x;

        printf("size of st1 = %d\n",sizeof(st1));
        printf("size of st2 = %d\n",sizeof(st2));

        for (x = 0; x < 3; x++)
        {
                printf("%s %d %d  %s %d %d\n",
                        st1[x].c1,st1[x].i1,st1[x].h1,
                        st2[x].c2,st2[x].i2,st2[x].h2);
```

McCosh C generates the following (incorrect) output:

```
size of st1 = 11
size of st2 = 15
 430 5  item21 1 2
 444 0  item22 3 4
 27002 101  item23 5 6
```

Introl C and UNIX C generate the following (correct) output:

```
size of st1 = 33
size of st2 = 15
item11 1 2  item21 1 2
item12 3 4  item22 3 4
item13 5 6  item23 5 6
```

James McCosh, the developer of the McCosh C compilers, is aware that this problem exists. The compiler incorrectly computes the size of the elements of and incorrectly initializes a structure containing character arrays. This may be corrected by now.

So that Introl will not be left out of the bug list, the following example, sent to me by Phil Gunsul, provides another problem, related to structures, in Introl v1.5:

```
#include    <stdio.h>

#define IFLEVEL 10
#define CLEVEL  10

struct indent {
        int tabs;
        int pdepth;
        int iflev;
        int ifc[IFLEVEL];
        int spdepth[IFLEVEL];
} ind[CLEVEL];
struct indent *clev = ind;

main()
{
        int x;

        clev->tabs = 5;
        clev->iflev = 3;
        clev->ifc[clev->iflev] = clev->tabs;
        clev->pdepth = clev->ifc[clev->iflev];
        printf("number = %d\n", clev->pdepth);
```

```
        clev->tabs = 0;
        clev->iflev = 4;
        z = clev->iflev;
        clev->ifc[z] = clev->tabs;
        clev->pdepth = clev->ifc[z];
        printf("number = %d\n", clev->pdepth);

        exit(0);
}
```

Introl C generates the following (incorrect) output:

```
number = 0
number = 8
```

McCosh C and UNIX C generate the following (correct) output:

```
number = 5
number = 8
```

HIDDEN LIMITATIONS

Many compilers, like other programs, have undocumented limitations which (hopefully) affect only worst-case and trivial cases.

Early versions of Introl restricted the link-edited program's to 32K, or more in some instances. The current version, being shipped since the first of the year, Ver. 1.5 - allows up to 64K.

The following example is taken from the May 1984 issue of BYTE. It tests the depth to which the unary pointer operator '*' may be nested, and demonstrates a problem with McCosh C.

```
#include "stdio.h"
#define LOOPS 50000
#define BELL  7
struct cptrl {
        char **********************ptr1;
};
main()
{
        unsigned i;
        char yeldorb;
        struct cptrl **********************pp:ster;
        printf ("Iu loops \n",LOOPS);
        for (i=0; i<=LOOPS; i++)
                yeldorb = **********************
                (**********************pointer).ptr1;
        printf ("%cfinished\n",BELL);
        exit(0);
}
```

Introl and UNIX C compile and run this example correctly, although McCosh C refuses to compile the statement within the "for" loop correctly. Admittedly, this is a contrived and worst-case example, but McCosh C should be able to compile and run it correctly, or document the fact that the compiler handles only five levels of pointers correctly.

DEVIATIONS FROM THE STANDARD

Some of the inconsistencies among the 6809 C compilers are due to deviations from the K & R and/or UNIX C standards. One problem with using the K & R book as a standard with which to develop compilers is that it is quite vague in many areas; simply stating that a compiler is compatible with K & R does not ensure that it can compile a large number of UNIX C programs. Another problem with referencing the K & R book is that it does not represent the latest version of UNIX C; such items as structure assignment, structure function arguments, 'enum' data type, and others are not included in K & R, since they were not implemented at that time.

A common problem which must be faced by someone attempting to port a C program from one compiler to another, even on the same machine, concerns the tremendous variation in the implementation and use of the "#include" preprocessor statement and the organization of the directories. Often, the correct combination of quotes, brackets, braces, etc. must be determined experimentally, as must be the correct library names and location of the libraries. Some of the compilers insist on using specific drives (in FLEX) or the boot drive (in OS/9), all of which is irritating to the user of a system with hard disks.

Introl C does not currently support doubles, and documents that it does not do so. It does support longs and floats, but the range of longs is limited and the precision of floats is limited. McCosh C supports longs, floats, and doubles.

Introl C supports nested comments, by default. Since UNIX C and McCosh C do not support nested comments, none should be used, and the option should be set to prohibit them, if compatibility is a concern.

McCosh C has its own deviation problems. It does not support line continuation with reverse slash and carriage return, as supported by Introl C and UNIX C. It has very strict requirements on external variable declarations not shared by Introl and UNIX C.

The toupper and tolower macros in the FLEX and UniFLEX versions of McCosh C fail to check for their arguments being within range before converting them to upper or lower case letters. These very same functions in Microware C do check their arguments in an identical manner to that performed by the current version of the true AT&T UNIX C (although not by some of the clones). UNIX C and Microware C change the names of the toupper and tolower macros to _toupper and _tolower, then declare functions toupper and tolower which check their arguments. This avoids the problem of repeating an argument within a a macro, thus causing double evaluation an strange side effects with expressions such as "*c++".

The following program illustrates this portability problem:

```
#include "stdio.h"
#include "ctype.h" /* omit for non-McCosh C */

char z;

main()
{
        for (z = '0'; z < 0x7f; z++)
                printf("%c, %c, %c\n",z,toupper(z),tolower(z));
}
```

McCosh C for FLEX and UniFLEX produces the following (incorrect) output.

```
0, , 0
9, , 9
:, , :
<, , <
=, , =
>, , >
?, , ?
@, @,
A, A, a
Z, Z, z
[, [, {
, , |
```

```
), ), )
^, ^, ^
-, -, -
/, /, /
0, 0, 0
A, A, A
Z, Z, Z
[, [, [
\, \, \
], ], ]
^, ^, ^
```

Introl C, UNIX C, and Microware C generate the following (correct) output.

```
0, 0, 0
9, 9, 9
:, :, :
<, <, <
=, =, =
>, >, >
?, ?, ?
@, @, @
A, A, A
Z, Z, Z
[, [, [
\, \, \
], ], ]
^, ^, ^
_, _, _
`, `, `
a, a, a
z, z, z
{, {, {
|, |, |
}, }, }
~, ~, ~
```

COMMON SURPRISE

One situation, which must be described as a surprise, rather than a bug or a deviation from standards, concerns the order of evaluation of function arguments. Although the K & R C book and the manuals for the various Full C compilers seem to imply that function arguments are evaluated in first to last order, UNIX C, Introl C, and McCosh C all evaluate their function arguments in reverse order. They do this so that the stack will be established with the first argument last in the stack. This simplifies the processing of functions such as printf and scanf which have a variable number of arguments. Most small C compilers and certain other Full C compilers evaluate function arguments in normal order.

The following program may be used to determine the order of evaluation in a given C compiler:

```
#include "stdio.h"
main()
{
        int i = 0;
        printf("%d %d %d %d %d",
                ++i,++i,++i,++i,++i);
}
```

Those compilers which evaluate function arguments in reverse order will output the following line:

```
5 4 3 2 1
```

Those compilers which evaluate function arguments in normal order will output the following line:

```
1 2 3 4 5
```

Although no change is required in the compilers, this unexpected evaluation order should at least be documented in the language manuals.

EDITOR'S NOTE:

The 'surprise' of first to last and last to first is partially explained in K&R, page 186 Appendix A, "The order of evaluation of arguments is undefined by the language; take note that the various compiler differ."

SUMMARY

The problems and situations described above are representative of the condition of the Full C compilers available on the 6809. Other readers are invited to send small, documented test cases on disk to me, after sending the problems to the developers. Perhaps, in this manner, the compiler manufacturers are alerted to other 'bugs' and manuals can be improved and corrected.

EXAMPLE PROGRAM

Following is this month's example C program; it performs two-up listings, and was provided by a reader. It will run under Introl v1.4 or v1.5 C, but the continued lines must be split up for McCosh C.

```
#include "stdio.h"

#define  MAXFILE  25      /* max number of input files */
#define  HELPMSG1 \
  "\7Usage: LONGSIDE ([file] [(width)] [(spaces)]),\n"
#define  HELPMSG2 "If (width) is null or zero, the whole \
input line is sent to output.\nFor (spaces) to be recognized, \
(width) must be given explicitly, even if zero.\n"
#define  HELPMSG3 "The first numeric argument after an \
alphabetic is taken as (width), the last\nas (spaces).  There \
is no way to specify leading spaces.\n"

char argnr,                 /* argument index */
  done,                     /* flag signaling completion */
  filenr,                   /* input file index */
  line[MAXLINE],            /* input buffer */
  nfiles,                   /* input file count */
  *s,                       /* character pointer */
  spaces[MAXFILE],          /* spaces to be output after line */
  w,                        /* width (temporary) */
  wflg[MAXFILE],            /* flag signaling width limit */
  width[MAXFILE];           /* width */
FILE *infp[MAXFILE],        /* file pointer */
  *fopen();

main(argc, argv)
int argc;
char *argv[]; {

/* give instructions if call lacks arguments */
  if (argc < 2) {
    fprintf(STDERR, HELPMSG1); fprintf(STDERR, HELPMSG2);
    fprintf(STDERR, HELPMSG3);
    exit();
  }

/* process command line and open input files */
  for (argnr = 1; argnr < argc && *argv[argnr]; ++argnr) {
    if (isalpha(*argv[argnr])) {
                /* an alpha argument can be only a filename */
      ++filenr;
      if ((infp[filenr] = fopen(argv[argnr], "r")) == ERROR) {
        fprintf(STDERR, "\7Can't open %s.\n", argv[argnr]);
        exit();
      }
      wflg[filenr] = NO;        /* start with no width limit */
       /* if arg is numeric and filename but no width is set,
                                    arg must be width */
else if (isdigit(*argv[argnr]) && infp[filenr])
  if (!wflg[filenr]) {
    width[filenr] = atoi(argv[argnr]);
    wflg[filenr] = YES;
  }
        /* if width set, any other numeric arg is spaces */
  else spaces[filenr] = atoi(argv[argnr]);
```

```
                        /* no other kind of argument allowed */
    else {
      fprintf(STDERR, "\7Wrong syntax.\n\n", HELPMSG);
      exit(1);
    }
  }
  nfiles = filenr;

/* for each line, read a line from each input file in turn */
  do {                  /* continue until all files finished */
    done = YES;             /* start by assuming we're done */
    for (filenr = 0; filenr < nfiles; ++filenr) {
      n = width[filenr] ? width[filenr] : 1;
      if (s = fgets(line, MAXLINE - 1, infp[filenr])) {
        done = NO;          /* not done if any input left */
        while (n && *s && *s != '\n') {
          putchar(*s++);        /* fill to width or end of line */
          if (width[filenr]) n--;   /* count only if width set */
        }
      }
            /* now pad with spaces and add separating spaces */
      if (width[filenr]) while (n--) putchar(' ');
      for (n = spaces[filenr]; n; n--) putchar(' ');
    }
    if (!done) putchar('\n');  /* newline after all files read */
  } while (!done);

}
```

CAT Utility for FLEX 1 2&9

In the August 1983 issue of '68 Journal there appeared a letter from Bud Pass mentioning my name as the author of the new utilities COPY and CAT. As always, the Journal brought quick response from several readers. Unfortunately my address did not appear, which must have stumped many readers who wanted to write but were unable to locate me.

Please publish my address in the next issue for the convenience of your readers. Better still, publish the COPY article I sent you a few months ago! COPY has been in use on many FLEX systems with only one system compatibility problem. It seems Data-Comp FLEX for the Color Computer will crash if a program alters the interrupt mask to pause the TSC printer spooler. Data-Comp users ONLY should remove the instructions SEI and CLI from the source of COPY.

I've enclosed the CAT program mentioned by Bud Pass. CAT was written to cram the most features into the limited FLEX utility transient area. It has all the features of the TSC commands CAT and DIR. Our program also has several unique ideas, such as the automatic computation of the maximum number of columns that will fit on the display and printer. This will be especially useful for CoCo users who usually display 51 characters per line, but have printers that are 80 or 132 columns wide.

I have been sending out disks for two years, and will continue to do so, but it has been quite an undertaking. I receive a wide variety of packages in the mail. If a reader would like a copy of the programs we are distributing, please send either 1 eight inch disk or 3 five inch disks in an envelope large enough to contain the instruction sheets that we've written for the larger programs (CAT, COPY, PROMPROG and RESIDENT). The disk mailing cartons many people send are too small for the literature. Overseas readers (of which there are many!) need only send a letter. The disks arrive in a variety of formats, some of which my system can't handle! I ALWAYS re-initialize them with the TSC standard SSSD 35/77 track format.

One final note: Bud Pass didn't mention the co-author of COPY and CAT. Bruno Puglia and I wrote these programs as a team.

Leo Taylor
109 Twin Brook Road
Hamden, Conn. 06514

A NEW CAT UTILITY FOR FLEX 1, 2 and 09

Bruno D. Puglia and Leo E. Taylor

START BY ASSEMBLING THE SOURCE

The user can assemble the CAT Program and run it on FLEX 1 or 2.0 (6800) or FLEX 09 (6809) systems. CAT is compatible with the old CAT but has features and options never available to the FLEX user. It can replace the old CAT, DIR and FILES programs.

Before doing the assembly, you can save your old CAT by renaming it CATOLD. You can now assemble the new CAT program. Since CAT is provided as a source file, you can assemble it with the TSC FLEX assembler for your 6800 or 6809 processor. You will have to change one letter in one line in order to select the type of processor used:

FLEX EQU $A000 for 6800 FLEX EQU $C000 for 6809

There are width equates which must be set for your display and printer. If these devices have auto return (CR), set the width count one less than the maximum line length. An 80 character terminal with auto-return on the 80th character should be defaulted to 79 in the source.

COCO NOTE: For Color Computers with 51 characters per line the best results will be attained with TWIDTH set to 52.

There are other options to set at this time. The page length is set for 66 lines and this value will be correct for normal printers and 11 inch paper. The form-feed string is set to $0C, but it can be set to the user's character. An alternate string of two CRLFs is listed in the source to keep paging correct if the printer does not have real form feed capability.

INITIAL SURPRISES

CAT will accept commands in the same format as the original CAT, but its display format is done in multiple columns across the screen. Another surprise is yet to come! As you start to use the various options the number of columns across the display or printer will automatically change depending on what options you have selected.

OPTION LETTERS

There are several option letters supported in CAT which make it a very powerful utility and maybe a bit overwhelming at first. As you use CAT you will begin to learn what the options are and how to use them effectively. After you learn what each option letter does, you may elect to go back and select your own set of default options in the source. All options are entered on the command line. If you have trouble remembering what the letters do, the CAT help list can be displayed by entering a command of CAT +.

Entering a command line with options : CAT +SDF1 1

If an unimplemented option letter is used, CAT will display the HELP list. CAT uses positive option logic. Entering an option letter on the command line will turn the option on. All others are turned off so you must enter ALL options you want if you enter any at all.

DEFAULT OPTIONS

CAT has default options which are used whenever there are no option letters entered on the command line. The source has provisions for setting your own set of default options. The user can disable or enable an option by placing an FCB 0 (OFF) or FCB 1 (ON) in the source option list. After the assembly of CAT, the user can check the options by typing CAT +. CAT's HELP list will display which options are enabled with a YES in front of the option letter. In dealing with defaults, you must remember that if any option is entered on the command line all defaulted options are turned off and all desired options have to be entered.

CAT + will display CAT's HELP list :

- Display -		- Comments -
Example: CAT +HD 1 .CMD A8		typical command line
No	A Alpha	Alpha A,B,C type listing
No	D Date	Display file Date
No	F File #	Display DIR File number
No	M Maxi-DIR	DIR full listing
No	N Non-existant	Display Deleted files with (-)
No	P Paging	Paging with Printer column width
No	R Repeat	Repeat CAT as listed on command line
Yes	S Size	Display file sector Size

EXPLANATION OF OPTION LETTERS

A CAT IN ALPHANUMERICAL GROUPS

The "A" option will group the listing by the first letter. This simple form of alphabetical CAT must scan the entire directory for each letter and this will take time. We do not feel this option should be defaulted on. When the "A" option is used you cannot use any match strings. Additional information will be found in the FEATURES section of this manual.

D DISPLAY FILE DATE

The "D" option will display the file's creation date. Not all FLEX users use real dates when first booting up FLEX so if the month is zero or over 12, CAT will display the month as 'BAD'. Dates can be very useful with our programs so the user will find it advantageous to use current dates.

F LIST THE DIRECTORY FILE NUMBER

The file number option will list the actual directory number.

M MAXIMUM DIR TYPE LISTING

The "M" option turns CAT into a DIR type program. The "M" option will provide information on track-sector data and the protect file code information. The "M" option turns on the D,F and S options plus routines needed for the header, track and sector, and protect codes. You can still add option letters to enhance the DIR type listing.

N LIST NON-EXISTING FILES ON THE DISK (DELETED FILE IN THE DIR)

The N option was chosen as "NON EXISTING" because it will display deleted file entries that exist in the directory. The actual file may not exist on the disk, it may have been over-written when it was part of the list of free sectors. You cannot assume a deleted file is intact unless you know it was recently deleted and you have not written enough new files on the disk to reuse the deleted file's sectors. In the disk's directory a deleted file will have the first letter of the file name changed to a $FF. CAT will display the $FF as a dash (-) as the first character of the file name. It is not a good practice to use one or two character file names since the first letter is deleted from the name when a file is deleted. When the "N" option is combined with the "M" option the DIR type listing will get you the starting sector of the deleted files in the directory.

P PROVIDE A PAGED LISTING and USE PRINTER WIDTH EQUATE

The "P" option will enable a paging subroutine. It will place blank lines on the top and bottom of the page. This is handy when listing a disk with a large number of files. The printer width equate is picked up and used to calculate the number of columns allowed for each form of the printed listing.

R REPEAT OF CAT USING THE EXISTING COMMAND LINE

The "R" option allows CAT to re-start. A prompt will appear at the end of each CAT; enter "E" to exit to FLEX, or hit any other key to do another CAT as originally listed on the command line. This option will allow you to change disks and do a CAT without re-entering the data on the command line. If the "F" option is combined with FLEX's D.CMD, a disk file of several disks can be made. The user can select CAT options to control the type of information listed in the disk file.

S DISPLAY FILE SECTOR SIZE

The "S" option will enable the sector size of files to be displayed.

EXAMPLES

CAT	List all files on the ASN-ed work drive using the default options set up in the source.
CAT 0,1 C .BAS	List all files starting with the letter C and all .BAS files on drive 0. After drive zero is done list drive 1. Since no options were given defaulted options will be used.
CAT *	Display HELP list showing options and defaulted options with a "YES" next to option letter.
CAT +2 1	Display listing of the file names only for drive 1 in two columns. You may overide the automatic column count by entering an option of 1-9.
CAT +MA	Display a DIR-type listing including the file number, name.ext, size, starting and ending track and sectors, date and protect codes and in "A" alpha groups of A B C etc.
P CAT +MNP	Print a DIR-type directory including all deleted files and page the listing.
O CATFILE CAT +RSDF1	Make a diskfile "CATFILE" with Size, Date, File number of the working drive. The catalog will be one column wide. When a cat is complete, the E prompt is displayed. When any key, except the "E", is pressed the CAT will repeat. "F" would return to FLEX and close CATFILE. This file could then be edited or spooled to the printer using PIP or PRINT commands.
P CAT +ASDFT	Print Alpha grouped listing to printer with the file number, name.ext, size and date.

MEMORY USAGE

Before discussing the features in CAT please accept the fact that CAT was a large program to begin with and we were committed to writing a CAT program that would fit into the utility space provided for by FLEX. We have attempted to squeeze in as many features as possible in the limited memory space. One feature that did not fit was the Random file indicator found in the 6809 DIR utility. Alternate memory usage was discussed and we made CAT re-enterable so the program could be moved to other free RAM areas like $E800 ($6809) or $CD00-$DFFF ($6800).

SAMPLE CATALOG

CAT includes information not found in the original FLEX utility. The following sample catalog will be used as an example:

```
CAT +N C

DRIVE:2  VOLUME:SOURCE   4  CREATED:31-MAY-82

FILE#  NAME     TYPE BEGIN  END   SIZE   DATE      PRT

   2 COPY99  .TXT 29-00 31-0F  128  21-JUN-82 D
  19 COPYMAN9.TXT 12-01 45-0C   88  20-JUN-82
  20 CAT100  .TXT 41-01 44-0A   55  14-JUN-82

FILES=21  BIGGY=306  TOTAL=291/1020  FREE=111
```

CREATED: The date the disk was initialized.

FILE# Sequential number of the file in the directory. Because this catalog shows only files starting with "C" the file numbers shown indicate their position in the directory.

BEGIN/END Starting and ending disk sectors for file.

DATE The date file was created or edited.

PRT The three FLEX protect codes are indicated by letters. The catalog protect bit is not honored by this CAT.

C-catalog protect D-delete protect W-write protect

FILES= Total number of files in the disk directory.

BIGGY= Largest file on the disk.

TOTAL= Total number of sectors displayed followed by the total number of sectors used on the disk.

FREE= Sectors remaining for use.

FEATURES

A simple A B C type alpha type sort was included in CAT for those disks which are not already alphabetized. Since the scan process will go through the entire directory for each letter A thru Z, the "A" alpha option takes a great deal of time. It would not be wise to enable the "A" option when setting up CAT defaults unless you really need it and were ready to accept the extra time required. If alphabetizing is a major concern for the user, then use our COPY program which will do a full alphabetize when a disk is copied.

We did not pick our options letters with keyboard letter placement in mind but it so happens the commonly used option letters fall in a line on the keyboard. A S D F - ALPHA SIZE DATE and FILE number. You may find this handy when typing in the option letters.

We have elected to use the "*" sign for options since it is consistent with other FLEX programs. The addition of having HELP built into CAT is required when several option letters are used. Automatic selection of the number of columns for both the display terminal and the printer make this CAT unique.

CONCLUSION

The programmers who wrote CAT hope you will enjoy using this new utility. As far as we know CAT is bug free. It is possible that you may locate a bug or make improvements to our CAT. We hope you would inform us if you make any changes. As you look over the source and use CAT look beyond CAT and consider the compatibility aspects of FLEX. CAT was written to be compatible with all FLEX systems that use 6800 processors. By using standard FLEX vectors and adaptable 6800-6809 code we have produced a program which can be used by most FLEX users. There is no advantage to writing a major utility for just one type FLEX system. Optimizing a utility like CAT to a given processor gains little if we have to give up compatibility. If you're thinking about adapting this code to a 6809 only, forget it. CAT was written to be compatible for both the 6800 and the 6809 so please let it stay that way.

FLEX users cannot afford to be split up into small little system or processor groups. The FLEX system compatibility must be kept intact and should be utilized with more compatible programs like this CAT and our COPY. If you feel as we do, then speak out and say so! If you write programs, think about the compatibility.

This program come to you in source form at "NO-CHARGE" so feel free to pass them along to other FLEX users on the same "NO CHARGE" basis.

Bruno D. Puglia
27 Maitland Ave.
Randolph, Mass. 02368

Leo E. Taylor
109 Twin Brook Road
Hamden, Conn. 06514


```
4                * Improved CAT UTILITY for FLEX 1,2,9
5                *
6                * Program by Bruno Puglia and Leo Taylor
7                * 109 Twin Brook Rd.
8                * Hamden, Conn.  06514
9                *
10               * Calling format for HELP:
11               *
12               *  CAT *       note no parameters
13               *
14               * Calling format for using default options:
15               *
16               *  CAT         complete disk in sort drive.
17               *  CAT 1,2     two complete disks.
18               *  CAT AB .TXT files starting with AB then
19               *              files with TXT extension.
20               *
21               * Calling format for setting options:
22               *
23               *  CAT +DAP    complete disk with date,
24               *              alphabatical, and paging.
```

```
                        *
                        * CAT +M LET  maximum DIR options of
                        *             files starting with LET.
                        *
                        * Option letters:
                        *
                        *   A  Alphabetical order
                        *   D  include file Date
                        *   F  include File number
                        *   M  Maximum dir
                        *   N  include Non-existant files
                        *   P  Paging
                        *   R  Repeat
                        *   S  include file Size
                        *   1-9 coluen count
                        *
                        * Option M is used to invoke several options
                        * at once. When Maxi-cat is used the following
                        * information about each file will be displayed:
                        *
                        *  FILE, NAME, BEGIN, END, SIZE, DATE, PROTECT
                        *
                        *
                        * There are several options that must be edited
                        * in the source before assembly. All options
                        * are located at the beginning of the source
                        * code. The default options are selected by
                        * setting flags; FCB 1 is used for those options
                        * you want defaulted. Hardware options are set to
                        * match your terminal and printer.
                        *
                        * Commas and spaces can be used interchangeably.
                        *
                              PAG
                        ********************************************
                        *
                        * FLEX Address assignments:
                        *
                        * Change the next EQUATE for 6800/6809.
                        * Use $A000 for 6800 or $C000 for 6809.
                        *
A000                    FLEX    EQU     $A000     START OF FLEX

A840                    FCB     EQU     FLEX+$840

AC02                    TTYEOL  EQU     FLEX+$C02
AC0E                    DATE    EQU     FLEX+$C0E
AC0C                    ASADRV  EQU     FLEX+$C0C
AC11                    LSTTRM  EQU     FLEX+$C11
AC14                    BUFPNT  EQU     FLEX+$C14
AC22                    OUTSWT  EQU     FLEX+$C22

AD03                    WARMS   EQU     FLEX+$D03
AD15                    GETCHR  EQU     FLEX+$D15
AD18                    PUTCHR  EQU     FLEX+$D18
AD1E                    PSTRNG  EQU     FLEX+$D1E
AD21                    CLASS   EQU     FLEX+$D21
AD24                    PCRLF   EQU     FLEX+$D24
AD27                    WILCH   EQU     FLEX+$D27
AD2D                    GETFIL  EQU     FLEX+$D2D
AD39                    OUTDEC  EQU     FLEX+$D39
AD3C                    OUTHEX  EQU     FLEX+$D3C
AD3F                    RPTERR  EQU     FLEX+$D3F

B406                    FMS     EQU     FLEX+$1406


A100                          ORG     FLEX+$100
                              PAG
                        ********************************************
                        *
                        * CAT COMMAND UTILITY
                        *
A100 20  47             CAT     BRA     EXEC
A102 03                 VN      FCB     3



                        * USER HARDWARE OPTIONS
                        *
                        * SET TERMINAL AND PRINTER WIDTH TO THE
                        * MAXIMUM NUMBER OF CHARACTERS YOUR
                        * HARDWARE WILL DISPLAY WITHOUT ADVANCING
                        * TO THE NEXT LINE. SET PAGE LENGTH TO
                        * CONTROL THE NUMBER OF LINES TO PRINT
                        * BEFORE FORM FEED.
                        *
004F                    TWIDTH  EQU     79        TYPICAL 63 or 79
004F                    PWIDTH  EQU     79        TYPICAL 79 or 131
0042                    PAGLEN  EQU     66        LINES PER PAGE



                        * PRINTER FORM FEED STRING
                        * IF YOUR PRINTER DOES NOT HAVE
                        * A TOP OF PAGE COMMAND USE:
                        * FFEED FCB $D,$A,$9,$A,4
                        *
A103 0C 00 00 00        FFEED   FCB     $C,0,0,0,4
     A107 04
                              PAG

                        * DEFAULT OPTION BYTES WILL BE COPIED
                        * TO OPTION FLAGS WHEN INITIALIZED.
                        *
A108 00                 DEFALT  FCB     0         ALPHABETIZE
A109 00                         FCB     0         DATE
A10A 00                         FCB     0         FILE NUMBER
A10B 00                         FCB     0         MAXI-DIR
A10C 00                         FCB     0         NON-EXISTANT
A10D 00                         FCB     0         PAGING
A10E 00                         FCB     0         REPEAT
A10F 01                         FCB     1         SIZE



                        * OPTION LETTERS
                        *
A110 41                 LETTER  FCB     'A
A111 44                         FCB     'D
A112 46                         FCB     'F
A113 4D                         FCB     'M
A114 4E                         FCB     'N
A115 50                         FCB     'P
A116 52                         FCB     'R
A117 53                         FCB     'S



                        * OPTION FLAGS
                        *
A118 00                 ALPFLG  FCB     0
A119 00                 DATFLG  FCB     0
A11A 00                 FILFLG  FCB     0
A11B 00                 MAIFLG  FCB     0
A11C 00                 NONFLG  FCB     0
A11D 00                 PAGFLG  FCB     0
A11E 00                 RPTFLG  FCB     0
A11F 00                 SIZFLG  FCB     0

                  0000  OFFSET  EQU     LETTER-DEFALT
                              PAG
                        * WORKSPACE VARIABLES
                        *
A120 00 00 00 00        NAME2   FCB     0,0,0,0   DUMMY SPACE
A124 00 00 00 00        NAME    FCB     0,0,0,0   FILE NAME
A128 00 00 00 00                FCB     0,0,0,0
A12C 00 00 00           EXT     FCB     0,0,0     FILE EXTENSION
A12F 00                 ALLFLG  FCB     0         1=ALL FILES
A130 00                 ALLDRV  FCB     0         ALL DRIVES
A131 00                 LOOP    FCB     0
A132 00                 LINCNT  FCB     0         LINE COUNTER
A133 00                 COLCNT  FCB     0         COLUMN COUNTER
A134 00                 MAXCOL  FCB     0         MAXIMUM COLUMNS
A135 0000               POINTR  FDB     0
A137 0000               TEMP    FDB     0
A139 0000               DATPNT  FDB     0
A13B 0000               INDEX   FDB     0
A13D 0000               SECCNT  FDB     0         TOTAL SECTORS THIS CAT
```

```
173  A13F 0000          FILNUM  FDB   0         FILE NUMBER COUNT
174  A141 0000          FITOTAL FDB   0         TOTAL FILE
175  A143 0000          SECTRS  FDB   0         TOTAL SECTORS THIS DISK
176  A145 0000          LARGST  FDB   0         LARGEST FILE
177  A147 0000          BUFSAV  FDB   0         START OF PARAMETERS
178                     PAG

179                     ********************************************
180                     *
181                     * EXECUTIVE PROGRAM CALLS SUBROUTINES
182                     *
183  A149 8E  AC14      EXEC    LDX   BUFPNT
184  A14C BF  A147             STX   BUFSAV    KEEP FOR REPEAT
185  A14F 8E  A147     EXEC2   LDX   BUFSAV
186  A152 BF  AC14             STX   BUFPNT    RESTORE TO BEGINNING
187  A155 8D  16               BSR   INIT      SET UP
188  A157 8D  30               BSR   PARSE     READ COMMAND LINE
189  A159 7D  A11B             TST   MAINFLG   MAINFRM OPTIONS?
190  A15C 27  09               BEQ   NOTMAI
191  A15E 7C  A11A             INC   FILFLG
192  A161 7C  A119             INC   DATFLG
193  A164 7C  A11F             INC   SIZFLG
194  A167 8D  A20C     NOTMAI  JSR   COLUMN    SET COLUMNS ACROSS
195  A16A 7E  A245             JMP   DIRECT



197                     ********************************************
198                     *
199                     * INITIALIZE WORKSPACE
200                     *
201  A16D 8E  A100     INIT    LDX   #DEFALT
202  A170 E6  84       MOVDEF  LDAB  0,X       MOVE DEFAULTS
203  A172 E7  88 10            STAB  OFFSET+2,X
204  A175 30  01               INX
205  A177 8C  A110             CPX   #LETTER
206  A17A 26  F4               BNE   MOVDEF
207  A17C BE  A120             LDX   #NAME2
208  A17F 6F  84       CLRTMP  CLR   0,X       clear workspace
209  A181 30  01               INX
210  A183 8C  A147             CPX   #BUFSAV   LAST VARIABLE
211  A186 26  F7               BNE   CLRTMP
212  A188 39           RETRN2  RTS



214                     ********************************************
215                     *
216                     * PARSE COMMAND LINE FOR OPTIONS
217                     *
218                     * IF BAD OPTION FOUND DISPLAY HELP LIST
219                     *
220  A189 8E  AC14     PARSE   LDX   BUFPNT
221  A18C 30  1F               DEX
222  A18E A6  84               LDAA  0,X
223  A190 81  2B               CMPA  #'+       PLUS SIGN?
224  A192 27  10               BEQ   GOTPLS
225  A194 A6  01               LDAA  1,X
226  A196 81  2B               CMPA  #'+       PLUS SIGN?
227  A198 26  EE               BNE   RETRN2

228  A19A 8D  A027             JSR   NXTCH     SKIP OVER PLUS
229  A19D A6  02               LDAA  2,X
230  A19F 8D  A3C9             JSR   TSTTR2    TERM AFTER PLUS?
231  A1A2 27  30               BEQ   HELP
232  A1A4 8E  A118     GOTPLS  LDX   #ALPFLG
233  A1A7 6F  84       CLROPT  CLR   0,X       CLEAR ALL FLAGS
234  A1A9 30  01               INX
235  A1AB 8C  A120             CPX   #SIZIFLG+1
236  A1AE 26  F7               BNE   CLROPT
237  A1B0 8D  A027     OPTLOP  JSR   NXTCH     GET OPTION CHAR
238  A1B3 25  03               BCS   RETRN2    NON-ALPHANUMERIC
239  A1B5 81  39               CMPA  #'9
240  A1B7 22  07               BHI   NOTCOL
241  A1B9 80  30               SUBA  #'0       MAKE NEW
242  A1BB B7  A134             STAA  MAICOL    1-9 IS COLUMN
243  A1BE 20  F0               BRA   OPTLOP
244
245  A1C0 84  5F       NOTCOL  ANDA  #$5F      MAKE UPPER CASE
246  A1C2 8E  A10F             LDX   #LETTER-1 OPTION LETTER TABLE
247  A1C5 30  01       OPTL02  INX
248  A1C7 8C  A118             CPX   #ALPFLG
249  A1CA 27  08               BEQ   HELP      NOT FOUND?
250  A1CC A1  84               CMPA  0,X
251  A1CE 26  F5               BNE   OPTL02
252  A1D0 6C  08               INC   OFFSET,X  SET FLAG
253  A1D2 20  BC               BRA   OPTLOP



255                     * PRINT HELP LIST
256                     *
257  A1D4 8E  A699     HELP    LDX   #HLPMSG   HELP HEADER
258  A1D7 8D  AD1E             JSR   PSTRNG
259  A1DA 8E  A118             LDX   #ALPFLG
260  A1DD 9F  A137             STX   TEMP      OPTION FLAG POINTER
261  A1E0 8E  A684.            LDX   #OPTMSG
262  A1E3 9F  A135     HLPLOP  STX   POINTR    SAVE MESSAGE
263  A1E6 8E  A137             LDX   TEMP
264  A1E9 A6  84               LDAA  0,X       GET FLAG
265  A1EB 39  01               INX
266  A1EB 9F  A137             STX   TEMP
267  A1FD 8E  A693             LDX   #NOMSG
268  A1F3 4D                   TSTA            FLAG SET?
269  A1F4 27  03               BEQ   ISNO
270  A1F6 8E  A68B             LDX   #YESMSG   SWITCH TO YES
271  A1F9 8D  AD1E     ISNO    JSR   PSTRNG    OUTPUT YES OR NO
272  A1FC 8E  A135             LDX   POINTR    RESTORE MESSAGE
273  A1FF 8D  A524             JSR   PDATA
274  A202 30  01               INX             SKIP EOT
275  A204 8C  A6FF             CPX   #M.STOPT+1 END OF OPTIONS?
276  A207 26  DA               BNE   HLPLOP
277  A209 7E  A3B0     EXIT2   JMP   EXIT

279                     * COMPUTE NUMBER OF COLUMNS THAT WILL FIT
280                     * ON TERMINAL. IF THE COMMAND LINE DOES
281                     * NOT SPECIFY THE NUMBER OF COLUMNS THIS
282                     * ROUTINE WILL ADD UP THE CHARACTERS NEEDED
283                     * PER COLUMN AND DIVIDE INTO TERMINAL SIZE.
284                     *
285  A20C 70  A134     COLUMN  TST   MAICOL    SPECIFIED ALREADY?
286  A20F 26  33               BNE   RETRN1
287  A211 C6  00               LDAB  #33       WIDTH OF NAM.EXT
288  A213 70  A11A             TST   FILFLG
289  A216 27  02               BEQ   FILOFF
290  A218 CB  06               ADDB  #6        +WIDTH OF FILE NAM
291  A21A 7D  A11F     FILOFF  TST   SIZFLG
292  A21D 27  02               BEQ   SIZOFF
293  A21F CB  06               ADDB  #6        +WIDTH OF SIZE-1
294  A221 7D  A119     SIZOFF  TST   DATFLG
295  A224 27  02               BEQ   DATOFF
296  A226 CB  0A               ADDB  #10       +WIDTH OF DATE
297  A228 7D  A11B     DATOFF  TST   MAIFLG
298  A22B 27  02               BEQ   MAIOFF
299  A22D CB  11               ADDB  #17       +WIDTH OF BEGIN/END/PROT
300  A22F 86  4F       MAIOFF  LDAA  #TRMIDTH  WIDTH OF TERMINAL
301  A231 7D  A11B             TST   PAGFLG
302  A234 27  02               BEQ   NOPAG2
303  A236 86  4F               LDAA  #PRNIDTH  WIDTH OF PRINTER
304  A238 7A  A134     NOPAG2  DEC   MAICOL    START -1
305  A23B 7C  A134     FNDCOL  INC   MAICOL
306  A23E 34  04 A0E0          SUBA            COMPUTE COLUMNS
307  A242 24  F7               BCC   FNDCOL
308  A244 39           RETRN1  RTS



310                     * FIND DRIVE NUMBER
311                     *
312  A245 8E  AC14     DIRECT  LDX   BUFPNT     POINT TO BUFFER
313  A248 9F  A135             STX   POINTR    SAVE POS
314  A24A 8D  A024     DIRLOP  JSR   PCRLF     PRINT CR & LF
315  A24E 7D  A130             TST   ALLDRV    ALL DRIVES?
316  A251 27  14               BEQ   DRLPCD
317  A253 86  14       NXTDRV  LDAA  #20       FIND DRIVE FUNCTION
318  A255 8D  A51C             JSR   CALFNS
319  A258 26  AF               BNE   EXIT2
320  A25A 8E  A135             LDX   POINTR    RESTORE POINTER
321  A25D 8F  AC14             STX   BUFPNT
322  A260 A6  84               LDAA  0,X       GET CHARACTER
323  A262 87  AC11             STAA  LSTTRM    SET TERM
324  A265 20  63               BRA   RETD12
```

```
325
326  A267 BE  A135  CHKPER LDX   POINTR   GET POINTER
327  A26A A6  84            LDAA  0,X      GET CHARACTER
328  A26C 81  2E            CMPA  #'.      IS IT PERIOD?
329  A26E 27  09            BEQ   CHKLOP

330  A270 BD  AD21          JSR   CLASS    CLASSIFY IT
331  A273 25  04            BCS   CHKLOP   TERM?
332  A275 81  39            CMPA  #'9      IS IT NUMBER?
333  A277 23  33            BLS   GETDRV
334  A279 7D  A131  CHKLOP  TST   LOOP     LOOPING?
335  A27C 27  1F            BEQ   USEASN
336  A27E 7D  A11E          TST   RPTFLG
337  A281 27  B6    EXIT3   BEQ   EXIT2
338  A283 B6  FF            LDAA  #$FF
339  A285 B7  AC22          STAA  OUTSWT
340  A288 CE  A662          LDX   #RPTMSG
341  A28B BD  AD7E          JSR   PSTRNG   ASK REPEAT
342  A28E 7F  AC22          CLR   OUTSWT
343  A291 BD  AD75          JSR   GETCHR
344  A294 84  5F            ANDA  #$5F     LOWER CASE
345  A296 81  45            CMPA  #'E
346  A298 27  E7            BEQ   EXIT3
347  A29A 7E  A14F          JMP   EXEC2    START OVER
348
349  A29D CE  A840  USEASN  LDX   #FCB     POINT TO FCB
350  A2A0 B6  AC0C          LDAA  ASNDRV   GET DRIVE
351  A2A3 A7  03            STAA  3,X      SET IN FCB
352  A2A5 2A  23            BPL   GETD12
353  A2A7 B7  A130          STAA  ALLDRV   SET ALL DRIVES
354  A2AA 20  A7            BRA   NITDRV
355
356  A2AC BE  A135  GETDRV  LDX   POINTR   RESTORE POINTER
357  A2AF BF  AC14          STX   BUFPNT
358  A2B2 BD  AD27          JSR   NXTCH    GET NEXT CHAR
359  A2B5 84  07            ANDA  #7       MASK NUMBER
360  A2B7 CE  A840          LDX   #FCB     POINT TO FCB
361  A2BA A7  03            STAA  3,X      SAVE IN DRIVE
362  A2BC BD  AD27          JSR   NXTCH    GET NEXT CHAR
363  A2BF 25  03            BCS   GETDIR   TERM?
364  A2C1 7E  A3B6          JMP   SYNTAX   IF NOT, ERROR
365
366  A2C4 BE  AC14  GETDIR  LDX   BUFPNT
367  A2C7 BF  A135          STX   POINTR
368  A2CA 7D  A110  GETD12  TST   ALPFLG
369  A2CD 27  03            BEQ   NOTALP
370  A2CF BD  A5CF          JSR   ALPHA
371  A2D2 7F  A133  NOTALP  CLR   COLCNT   RESET COLUMNS
372  A2D5 B6  3B            LDAA  #PAGLEN-7
373  A2D7 B7  A132          STAA  LINCNT   RESET LINES
374  A2DA B6  10            LDAA  #16      OPEN SYSTEM REC
375  A2DC BD  A51C          JSR   CALFMS
376  A2DF 26  09            BNE   ERROR3
377  A2E1 7F  A13D          CLR   SEQCNT   RESET THIS CAT
378  A2E4 7F  A13E          CLR   SECCNT+1
379  A2E7 BD  A51A          JSR   GETIR    GET FIRST RECORD
380  A2EA 26  75    ERROR3  BNE   ERROR2
381  A2EC CE  A648          LDX   #DIRST   DRIVE:
382  A2EF BD  AD1E          JSR   PSTRNG
383  A2F2 CE  A842          LDX   #FCB+2   DRIVE

384  A2F5 6F  84            CLR   0,X      USE TWO BYTES
385  A2F7 BD  A5C2          JSR   OUTDE2
386  A2FA CE  A658          LDX   #VOLMSG  VOLUME:
387  A2FD BD  A524          JSR   PDATA
388  A300 CE  A844          LDX   #FCB+4   POINT TO VOLUME
389  A303 C6  08            LDAB  #8
390  A305 BD  A526          JSR   PDATAB
391  A308 BD  A515          JSR   OUTSPC
392  A30B CE  A84F          LDX   #FCB+15  POINT TO NUMBER
393  A30E BD  A3B0          JSR   DECSPC
394  A311 CE  A64F          LDX   #CREATN  CREATED:
395  A314 BD  A524          JSR   PDATA
396  A317 CE  A857          LDX   #FCB+23  POINT TO DATE
397  A31A BD  A586          JSR   OUTDAT
398  A31D BD  AD24          JSR   PCRLF
399  A320 7D  A116          TST   MAKFLG
400  A323 27  09            BEQ   NOMEAD
401  A325 CE  A5FA          LDX   #HEADER  FILE NAME ETC
402  A328 BD  AD1E          JSR   PSTRNG
403  A32B BD  AD24          JSR   PCRLF

404  A32E BD  A3C6  NOMEAD  JSR   TSTTRM
405  A331 26  05            BNE   CHKPR2
406  A333 7F  A12F          CLR   ALLFLG   CLEAR FLAG
407  A336 20  1E            BRA   MATCH2
408
409  A338 CE  AC14  CHKPR2  LDX   BUFPNT   RESET POINTER
410  A33B A6  84            LDAA  0,X      GET NEXT CHAR
411  A33D 81  2E            CMPA  #'.      IS IT PERIOD?
412  A33F 27  10            BEQ   MATCH
413  A341 BD  AD21          JSR   CLASS    CLASSIFY IT
414  A344 25  70            BCS   SYNTAX   TERM?
415  A346 81  39            CMPA  #'9      IS IT NUMBER?
416  A348 22  07            BHI   MATCH
417  A34A BD  AD27  ENDTRM  JSR   NXTCH    GET NEXT
418  A34D 24  F8            BCC   ENDTRM   TERM?
419  A34F 20  DD            BRA   NOMEAD
420
421  A351 BD  A53E  MATCH   JSR   GETNAM   INPUT NAME
422  A354 25  60            BCS   SYNTAX   ERROR?
423  A356 8D  79    MATCH2  BSR   PCAT     GO PRINT DIR
424  A358 8D  6C            BSR   TSTTRM   TEST TERM
425  A35A 26  F5            BNE   MATCH
426  A35C 86  10            LDAA  #16      OPEN SYSTEM REC
427  A35E BD  A51C          JSR   CALFMS
428  A361 26  5A    ERROR2  BNE   ERROR
429  A363 BD  A51A          JSR   GETIR    GET FIRST RECORD
430  A366 26  55            BNE   ERROR
431  A368 BD  AD24          JSR   PCRLF
432  A36B CE  A62D          LDX   #FILMSG  FILES=
433  A36E BD  AD1E          JSR   PSTRNG
434  A371 CE  A141          LDX   #FTOTAL
435  A374 BD  3A            BSR   DECSPC   TOTAL FILES
436  A376 CE  A63B          LDX   #LARMSG  #166Y=
437  A379 BD  A524          JSR   PDATA

438  A37C CE  A145          LDX   #LARGST
439  A37F BD  2F            BSR   DECSPC   PRINT LARGEST
440  A381 CE  A634          LDX   #SECMSG  TOTAL=
441  A384 BD  A524          JSR   PDATA
442  A387 CE  A130          LDX   #SECCNT
443  A38A BD  A5C2          JSR   OUTDE2   PRINT SECTORS THIS CAT
444  A38D 86  2F            LDAA  #'/
445  A38F BD  AD18          JSR   PUTCHR
446  A392 CE  A143          LDX   #SECTRS
447  A395 BD  19            BSR   DECSPC   PRINT SECTORS THIS DISK
448  A397 CE  A642          LDX   #FREEMS  FREE=
449  A39A BD  A524          JSR   PDATA
450  A39D CE  A855          LDX   #FCB+21
451  A3A0 BD  0E            BSR   DECSPC   PRINT AVAILABLE SECTORS
452  A3A2 7D  A11D          TST   PAGFLG
453  A3A5 27  06            BEQ   NOPAGE
454  A3A7 CE  A103          LDX   #FFEED   PAGE EJECT STRING
455  A3AA BD  AD1E          JSR   PSTRNG
456  A3AD 7E  A248  NOPAGE  JMP   DIRLOP   REPEAT
457
458  A3B0 BD  A5C2  DECSPC  JSR   OUTDE2   DECIMAL, THEN SPACES
459  A3B3 7E  A513          JMP   OUT2SP


461                    * REPORT ERROR AND EXIT
462                    *
463  A3B6 CE  AB40  SYNTAX LDX   #FCB      POINT TO FCB
464  A3B9 C6  1A           LDAB  #26       SET IN ERROR
465  A3BB E7  01           STAB  1,X
466  A3BD BD  AD3F  ERROR  JSR   RPTERR    REPORT ERROR
467  A3C0 BD  AD24  EXIT   JSR   PCRLF     PRINT CR & LF
468  A3C3 7E  A003         JMP   WARMS     RETURN TO FLEX


470                    * TEST FOR TERMINATOR
471                    *
472  A3C6 B6  AC11  TSTTRM LDAA  LSTTRM    CHECK LAST TERM
473  A3C9 81  0D    TSTTR2 CMPA  #$0D      IS IT CR?
474  A3CB 27  03           BEQ   RETRN3
475  A3CD 81  ACD2         CMPA  TTYEOL    IS IT EOL?
476  A3D0 39        RETRN3 RTS
```

```
478                     * PRINT DIRECTORY LIST
479                     *
480  A3D1 8E  A13F  PCAT  LDX   #FILNUM   CLEAR COUNTERS
481  A3D4 BD  A17F        JSR   CLRTMP
482  A3D7 86  06          LDAA  #6       OPEN DIRECTORY
483  A3D9 BD  A51C        JSR   CALFMS
484  A3DC 26  3F          BNE   ERROR
485  A3DE 7C  A131        INC   LOOP     SET LOOP FLAG

486  A3E1 BD  A51A  PCAT2 JSR   GETIR    GET INFO RECORD
487  A3E4 27  08          BEQ   PCAT3    ERRORS?
488  A3E6 A6  01          LDAA  1,X
489  A3E8 81  08          CMPA  #8       IS IT EOF?
490  A3EA 27  E4          BEQ   RETRKS
491  A3EC 20  CF          BRA   ERROR
492
493  A3EE 70  A844  PCAT3 TST   FCB+4     END OF DIR?
494  A3F1 27  DB          BEQ   RETRKS
495  A3F3 BE  A13F        LDX   FILNUM   BUMP FILE NUMBER
496  A3F6 30  01          INX
497  A3F8 BF  A13F        STX   FILNUM   SAVE NEW
498  A3FB 70  A844        TST   FCB+4
499  A3FE 2A  07          BPL   KEEP     DELETED FILE?
500  A400 70  A11C        TST   NOWFLG
501  A403 26  2E          BNE   KEEPDL   DELETED OK?
502  A405 20  0A          BRA   PCAT2    SKIP DELETED
503
504  A407 B6  A855  KEEP  LDAA  FCB+21   GET FILE SIZE
505  A40A F6  A856        LDAB  FCB+22
506  A40D B1  A145        CMPA  LARGST   LARGEST?
507  A410 25  00          BLO   NOLARG
508  A412 22  05          BHI   ISLARG
509  A414 F1  A146        CMPB  LARGST+1
510  A417 23  06          BLS   NOLARG
511  A419 B7  A145  ISLARG STAA  LARGST   SET NEW LARGEST
512  A41C F7  A146        STAB  LARGST+1
513  A41F FB  A144  NOLARG ADDB  SECTRS+1 ADD IN SECTOR COUNT
514  A422 89  A143        ADCA  SECTRS
515  A425 B7  A143        STAA  SECTRS   SAVE NEW COUNT
516  A428 F7  A144        STAB  SECTRS+1
517  A42B DE  A141        LDX   FTOTAL   BUMP FILE COUNT
518  A42E 30  01          INX
519  A430 BF  A141        STX   FTOTAL
520  A433 BE  A844  KEEPDL LDX  #FCB+4    POINT TO NAME
521  A436 70  A12F        TST   ALLFLG   ALL FILES?
522  A439 26  29          BNE   PCAT4
523  A43B BF  A139        STX   DATPNT   SAVE POINTER
524  A43E 8E  A124        LDX   #NAME    TRY TO MATCH NAME
525  A441 BF  A13B        STX   INDEX
526  A444 BE  A13B  PCAT5 LDX   INDEX
527  A447 8C  A12F        CPX   #EOIT+3
528  A44A 27  1B          BEQ   PCAT4
529  A44C A6  84          LDAA  0,X
530  A44E 30  01          INX
531  A450 BF  A13B        STX   INDEX
532  A453 BE  A139        LDX   DATPNT
533  A456 4D              TSTA           CHAR NULL?
534  A457 27  04          BEQ   ISNULL
535  A459 A1  84          CMPA  0,X      NO COMPARISON
536  A45B 26  84          BNE   PCAT2
537  A45D 30  01  ISNULL  INX            MOVE TO NEXT
538  A45F BF  A139        STX   DATPNT
539  A462 20  E0          BRA   PCAT5

540
541  A464 B6  A855  PCAT4 LDAA  FCB+21   ADD SIZE TO COUNT
542  A467 F6  A856        LDAB  FCB+22
543  A46A FB  A13E        ADDB  SECCNT+1
544  A46D 89  A13D        ADCA  SECCNT
545  A470 F7  A13E        STAB  SECCNT+1
546  A473 B7  A13D        STAA  SECCNT
547  A476 7D  A133        TST   COLCNT   LINE FULL?
548  A479 26  1E          BNE   NOCRLF
549  A47B B6  6134        LDAA  #MAICOL  RESET MAXIMUM ACROSS
550  A47E B7  A133        STAA  COLCNT
551  A481 BD  AB24        JSR   PCRLF
552  A484 7D  A11B        TST   PAGFLG   PAGING?
553  A487 27  10          BEQ   NOCRLF
554  A489 7A  A132        DEC   LINCNT
555  A48C 26  08          BNE   NOCRLF
556  A48E 8E  A103        LDX   #OFFEED
```

```
557  A491 BD  AD1E        JSR   PSTRNG   EJECT PAGE
558  A494 86  3F          LDAA  #PAGLEN-3
559  A496 B7  A132        STAA  LINCNT
560  A499 7A  A133  NOCRLF DEC  COLCNT
561  A49C 7D  A11A        TST   FILFLG
562  A49F 27  0A          BEQ   NOFILE
563  A4A1 C6  01          LDAB  #1       SET FLAG
564  A4A3 8E  A13F        LDX   #FILNUM
565  A4A6 BD  AB39        JSR   OUTDEC   PRINT FILE NUMBER
566  A4A9 8D  6A          BSR   OUTSPC
567  A4AB 8E  A844  NOFILE LDX  #FCB+4    POINT TO NAME
568  A4AE C6  08          LDAB  #8
569  A4B0 8D  74          BSR   PDATAB   PRINT NAME
570  A4B2 86  2E          LDAA  #'.
571  A4B4 BD  AD1D        JSR   PUTCHR   OUTPUT PERIOD
572  A4B7 C6  03          LDAB  #3
573  A4B9 8D  6B          BSR   PDATAB   PRINT EXT
574  A4BB 7D  A11B        TST   MAIFLG
575  A4BE 26  05          BNE   DOSPC
576  A4C0 7D  A11F        TST   SIZFLG
577  A4C3 26  02          BNE   SKPSPC   IF SIZE FOLLOWS NAME
578  A4C5 8D  4E  DOSPC   BSR   OUTSPC
579  A4C7 7D  A11B  SKPSPC TST  MAIFLG
580  A4CA 27  0E          BEQ   NOTMI2
581  A4CC 8E  A851        LDX   #FCB+17  OUTPUT START
582  A4CF BD  A5C6        JSR   OUTPNT
583  A4D2 8D  41          BSR   OUTSPC
584  A4D4 8E  A853        LDX   #FCB+19  OUTPUT END
585  A4D7 8D  5C          JSR   OUTPNT
586  A4DA 7D  A11F  NOTMI2 TST  SIZFLG
587  A4DD 27  0A          BEQ   NOTSIZ
588  A4DF 8E  A855        LDX   #FCB+21
589  A4E2 C6  01          LDAB  #1
590  A4E4 8D  AB39        JSR   OUTDEC   OUTPUT SIZE
591  A4E7 8D  2A          BSR   OUT2SP
592  A4E9 7D  A119  NOTSIZ TST  DATFLG
593  A4EC 27  0D          BEQ   NOTDAT

594  A4EE 8E  A859        LDX   #FCB+25
595  A4F1 8D  A58A        JSR   OUTDAT   OUTPUT DATE
596  A4F4 8D  1F          BSR   OUTSPC
597  A4F6 7D  A11B  NOTDAT TST  MAIFLG
598  A4F9 27  15          BEQ   NOTMI3
599  A4FB 8E  A840        LDX   #FCB     POINT TO FCB
600  A4FE 86  57          LDAA  #'W      OUTPUT PROT CODES
601  A500 C6  80          LDAB  #$80     WRITE?
602  A502 8D  7C          BSR   OUTPRO
603  A504 86  44          LDAA  #'D
604  A506 C6  40          LDAB  #$40     DELETE?
605  A508 8D  76          BSR   OUTPRO
606  A50A 86  43          LDAA  #'C
607  A50C C6  10          LDAB  #$10     CATALOG?
608  A50E 8D  70          BSR   OUTPRO
609  A510 7E  A3E1  NOTMI3 JMP  PCAT2    REPEAT



611                     * OUTPUT SPACE
612                     *
613  A513 8D  00  OUT2SP BSR   OUTSPC
614  A515 86  20  OUTSPC LDAA  #$20      SPACE
615  A517 7E  AD1D  PUTCH2 JMP  PUTCHR   PRINT IT


617                     * GET INFO RECORD
618                     *
619  A51A 86  07  GETIR  LDAA  #7        GET IR CODE
620  A51C 8E  A840  CALFMS LDX  #FCB     POINT TO FCB
621  A51F A7  04          STAA  0,X
622  A521 7E  B406        JMP   FMS      CALL FMS


624                     * PRINT DATA UNTIL EOT or
625                     * PRINT DATA UNTIL B=0
626                     *
627  A524 C6  FF  PDATA  LDAB  #$FF      USE EOT
628  A526 A6  84  PDATAB LDAA  0,X       GET CHARACTER
629  A528 26  02          BNE   PDATA2   NULL?
630  A52A 86  20          LDAA  #$20     SPACE
```

```
631  A52C 81 04      PDATA2  CMPA  #4
632  A52E 27 0D              BEQ   RETRN4
633  A530 81 FF              CMPA  #$FF      DELETED FILE
634  A532 26 02              BNE   NOTDEL
635  A534 86 20              LDAA  #' -
636  A536 BD DF      NOTDEL  BSR   PUTCH2    PRINT CHAR
637  A538 30 01              INX             BUMP TO NEXT
638  A53A 5A                 DECB
639  A53B 26 E9              BNE   PDATAB
640  A53D 39         RETRN4  RTS

642                     * GET FILE MATCH NAME
643                     *
644  A53E 7F A12F    GETNAM  CLR   ALLFLG    CLEAR ALL
645  A541 8E A120            LDS   #NAME2    POINT TO BUFFER
646  A544 C6 0F              LDAB  #15       SET COUNT
647  A546 6F 84      GETNA2  CLR   0,X       CLEAR BUFFER
648  A548 30 01              INX
649  A54A 5A                 DECB
650  A54B 26 F9              BNE   GETNA2
651  A54D BE AC14            LDS   BUFPNT    POINT TO NAME
652  A550 A6 84              LDAA  0,X       GET CHARACTER
653  A552 81 2E              CMPA  #'.       IS IT '.'?
654  A554 27 06              BEQ   GETNA4
655  A556 8E A120            LDS   #NAME2    RESET POINTER
656  A559 7E A02D            JMP   GETFIL    GET FILE NAME
657
658  A55C 7C AC15    GETNA4  INC   BUFPNT+1  FIX POINTER
659  A55F C6 03              LDAB  #3        SET COUNT
660  A561 8E A12C            LDS   #EXT      POINT TO EXT
661  A564 BD AD27    GETNA5  JSR   KITCH     GET CHARACTER
662  A567 25 10              BCS   GETNA6    TERM?
663  A569 81 60              CMPA  #$60      LOWER CASE?
664  A56B 25 02              BLO   NOTLOW
665  A56D 84 5F              ANDA  #$5F
666  A56F A7 84      NOTLOW  STAA  0,X       SAVE CHAR
667  A571 30 01              INX
668  A573 5A                 DECB
669  A574 2A EE              BPL   GETNA5
670  A576 1A 01              SEC
671  A578 39                 RTS             RETURN
672
673  A579 1C FE      GETNA6  CLC
674  A57B 39                 RTS



676                     * PRINT MINUS SIGN
677                     *
678  A57C 86 2D      OUTDSH  LDAA  #' -      DASH
679  A57E 20 97              BRA   PUTCH2    PRINT IT


681                     * OUTPUT PROT CODE
682                     *
683  A580 E5 0F      OUTPRO  BITB  15,X      TEST BYTE
684  A582 26 93              BNE   PUTCH2
685  A584 20 0F              BRA   OUTSPC    PRINT SPACE


687                     * OUTPUT DATE
688                     *
689  A586 8F A137    OUTDAT  STS   TEMP      SAVE POINTER
690  A589 A6 01              LDAA  1,X       GET DAY
691  A58B BD 29              BSR   OUTB      PRINT IT
692  A58D BD ED              BSR   OUTDSH
693  A58F BE A137            LDS   TEMP
694  A592 A6 84              LDAA  0,X       GET MONTH
695  A594 27 04              BEQ   BADMON
696  A596 81 0D              CMPA  #13
697  A598 25 02              BLO   OKMON
698  A59A 86 00      BADMON  LDAA  #13
699  A59C BE A6AA    OKMON   LDS   #MONTHS   POINT TO MONTHS
700  A59F 4A         OUTDA4  DECA
701  A5A0 27 08              BEQ   OUTDA5    FIND IN TABLE
702  A5A2 30 01              INX
703  A5A4 30 01              INX
704  A5A6 30 01              INX
705  A5A8 20 F5              BRA   OUTDA4
706
707  A5AA C6 03      OUTDA5  LDAB  #3        SET COUNT
708  A5AC BD A526            JSR   PDATAB    PRINT MONTH
709  A5AF BD CB              BSR   OUTDSH
710  A5B1 BE A137            LDS   TEMP      RESET POINTER
711  A5B4 A6 02              LDAA  2,X       GET YEAR
712  A5B6 BE A120    OUTD    LDS   #NAME2    USE FOR DIGIT
713  A5B9 A7 01              STAA  1,X       SAVE CHAR
714  A5BB 91 09              CMPA  #9        LESS THAN 10?
715  A5BD 22 03              BHI   OUTDE2
716  A5BF BD A515            JSR   OUTSPC
717  A5C2 5F         OUTDE2  CLRB
718  A5C3 7E AD39            JMP   OUTDEC    PRINT DIGITS



720                     * PRINT DISK ADDRESS
721                     *
722  A5C6 BD 04      OUTPNT  BSR   OUTHI2    OUTPUT DIGIT
723  A5C8 BD B2              BSR   OUTDSH    PRINT '-'
724  A5CA 30 01              INX
725  A5CC 7E AD3C    OUTHI2  JMP   OUTHEX    PRINT REST



727                     * FILL LINE BUFFER WITH ALPHABET
728                     *
729  A5CF BE AC14    ALPHA   LDS   BUFPNT
730  A5D2 C6 20              LDAB  #$20      SPACE
731  A5D4 F7 AC11            STAB  LSTTRM    DESTROY
732  A5D7 A6 84              LDAA  0,X
733  A5D9 81 00              CMPA  #$0D
734  A5DB 26 07              BNE   NOTCR
735  A5DD E7 84              STAB  0,X
736  A5DF 30 01              INX
737  A5E1 BF AC14            STS   BUFPNT
738  A5E4 86 41      NOTCR   LDAA  #'A       FIRST LETTER

739  A5E6 A7 84      ALPHA2  STAA  0,X       STUFF LETTER
740  A5E8 30 01              INX
741  A5EA E7 84              STAB  0,X       STUFF SPACE
742  A5EC 30 01              INX
743  A5EE 4C                 INCA
744  A5EF B1 5A              CMPA  #'Z
745  A5F1 25 F3              BLO   ALPHA2
746  A5F3 30 1F              DEX
747  A5F5 86 0D              LDAA  #$0D
748  A5F7 A7 84              STAA  0,X
749  A5F9 39                 RTS



751                     * MESSAGES
752                     *
753  A5FA 46 49 4C 45 HEADER  FCC   'FILE# NAME   TYPE BEGIN
     A5FE 23 20 20 4E
     A602 41 4D 45 20
     A606 20 20 54 59
     A60A 50 45 20 42
     A60E 45 47 49 4E
     A612 20 20
754  A614 45 4E 44 20         FCC   'END  SIZE  DATE   PRT'
     A618 20 20 53 49
     A61C 5A 45 20 20
     A620 20 44 41 54
     A624 45 20 20 20
     A628 20 50 52 54
755  A62C 04                  FCB   4
756  A62D 46 49 4C 45 FILMSG  FCC   'FILES='
     A631 53 3D
757  A633 04                  FCB   4
758  A634 54 4F 54 41 SECMSG  FCC   'TOTAL='
     A638 4C 3D
759  A63A 04                  FCB   4
760  A63B 42 49 47 47 LARMSG  FCC   'BIGGY='
     A63F 59 3D
761  A641 04                  FCB   4
762  A642 46 52 45 45 FREEMS  FCC   'FREE='
     A646 3D
```

```
763   A647 04                        FCB   4
764   A648 44 52 49 56  DIRST   FCC   'DRIVE:'
      A64C 45 3A
765   A64E 04                        FCB   4
766   A64F 43 52 45 41  CREATM  FCC   'CREATED:'
      A653 54 45 44 3A
767   A657 04                        FCB   4
768   A658 20 20 56 4F  VOLMSG  FCC   '  VOLUME:'
      A65C 4C 55 4D 45
      A660 3A
769   A661 04                        FCB   4
770   A662 2D 45 3F     RPTMSG  FCC   '-E?'
771   A665 04                        FCB   4

772   A666 4A 41 4E 46  MONTHS  FCC   'JANFEBMARAPRMAYJUN'
      A66A 45 42 4D 41
      A66E 52 41 50 52
      A672 4D 41 59 4A
      A676 55 4E
773   A678 4A 55 4C 41                FCC   'JULAUGSEPOCTNOVDEC'
      A67C 55 47 53 45
      A680 50 4F 43 54
      A684 4E 4F 56 44
      A688 45 43
774   A68A 42 41 44                   FCC   'BAD'
775
776   A68D 59 45 53 20  YESMSG  FCC   'YES '
      A691 20
777   A692 04                        FCB   4
778   A693 20 4E 4F 20  NOMSG   FCC   ' NO '
      A697 20
779   A698 04                        FCB   4
780   A699 45 78 61 6D  HLPMSG  FCC   'Example: CAT '
      A69D 70 6C 65 3A
      A6A1 20 43 41 54
      A6A5 20
781   A6A6 2B 04 46 20                FCC   '+DF 1 .CMD A8'
      A6AA 31 20 2E 43
      A6AE 4D 44 20 41
      A6B2 42
782   A6B3 04                        FCB   4
783   A6B4 41 20 41 6C  OPTMSG  FCC   'A Alpha'
      A6B8 70 68 61
784   A6BB 04                        FCB   4
785   A6BC 44 20 44 61                FCC   'D Date'
      A6C0 74 65
786   A6C2 04                        FCB   4
787   A6C3 46 20 46 69                FCC   'F File #'
      A6C7 6C 65 20 23
788   A6CB 04                        FCB   4
789   A6CC 4D 20 4D 61                FCC   'M Maxi-DIR'
      A6D0 78 69 2D 44
      A6D4 49 52
790   A6D6 04                        FCB   4
791   A6D7 4E 20 4E 6F                FCC   'N Non-existant'
      A6DB 6E 2D 65 78
      A6DF 69 73 74 61
      A6E3 6E 74
792   A6E5 04                        FCB   4
793   A6E6 50 20 50 61                FCC   'P Paging'
      A6EA 67 69 6E 67
794   A6EE 04                        FCB   4
795   A6EF 52 20 52 65                FCC   'R Repeat'
      A6F3 70 65 61 74
796   A6F7 04                        FCB   4
797   A6F8 53 20 53 69                FCC   'S Size'
      A6FC 7A 65
798   A6FE 04           LSTOPT  FCB   4
799
800                             END   CAT

0 ERROR(S) DETECTED
```

SUPPORT YOUR ADVERTISERS

LETNUM

Geoffrey A. Gass
5240 S.W. Doach Rd.,
Portland, Oregon 97201

LETNUM Routine Does
Quick Classification

The CLASS subroutine in FLEX spends several microseconds to determine for the calling program whether or not an ASCII character in Accumulator A is alphanumeric or not. Similar routines can be found in many high-level languages, compilers, assemblers and system software utilities.

Unfortunately, many or even most of the calling programs must spend nearly as much time again determining just what kind of alphanumeric or non-alphanumeric the character is before finally disposing of it.

A better classification routine segregates letters, numbers, punctuation and control characters, returning the information to the calling program via the flags available in the 68XX Condition Code Register. The calling routine need only invoke appropriate conditional branches to deal with the information efficiently.

The routine I have been using as a standard module in machine language programs is called LETNUM.

Because not all 16 possible combinations of the four convenient flags (N, Z, V, C) can be tested by single conditional branch instructions, and because some combinations are awkward to establish, the flags for character classification were set up rather arbitrarily, but I have not found any way to improve on them significantly.

Z — Set for any control character $00 - $1F. Usually signifies "termination". Tested with BEQ or BNE.

V — Set for a space, or any punctuation or symbol character $20-$2F, $3A-$40, $5B-$5F or $7B-$7F. Usually signifies "delimiter" (end of a word or number group). Tested with BVS or BVC.

N or C — Set for any number $30 - $39. Note: in this routine, N and C are set together and cleared together. Test with BMI, BPL, BCS or BCC. If Z, V, N and C are clear, the character is a letter.

The combined condition code tests may be profitably used in some cases:

BGT (Z clear; N ⊕ V): After LETNUM, V and N are never both set at the same time. Therefore BGT yields a branch if A contains a letter.

BLE (Z or (N exor V) set): Branch if A does not contain a letter.

BHI (Z clear and C clear): Branch if A contains a letter or delimiter (punctuation etc.).

BLT (V exor N set): Branch if A contains a number or a delimiter.

BGE (V ⊕ N): V and N cannot both be high after LETNUM. Branch if A contains a letter or a terminator (control character).

BLS (Z or C set): Branch if A contains a number or a terminator.

One important combined test <u>not</u> implemented by LETNUM is the alphanumeric/non-alphanumeric test. If this is the only distinction required, and saving a double branch is important, then call CLASS instead of LETNUM.

The various flags have different degrees of persistence after execution of various subsequent instructions. In the 6800, the least persistent will be the Z flag, which may be changed by any loading, storing or change in value of an accumulator or the

index register. The N and V flags are slightly more persistent, and are unchanged by an increment or decrement of the index register. The C flag is quite persistent, and is retained through incrementing and decrementing the accumulators, index register comparisons and even BIT (bit test) instructions. C is cleared by any TST or CLR instruction, and may be changed by any ADD, SUB or CMP instruction (except CPX). None of the flags are affected by PSH, PUL, RTS or branch instructions.

As a general rule, all testing should be complete as soon as possible after the return from LETNUM -- particularly if one of the combination tests (BGT, BGE, etc.) is to be employed.

Next time you're putting together a command-parser or type-checking routines, LETNUM may let you consolidate a lot of testing code into a single subroutine. Give it a try!

```
PAGE  001   LETNUM


0001Ø                    NAM   LETNUM   Character Sorting Routine
0002Ø                *         Rev 0.2  August 17, 1982
0003Ø                    OPT   O,NOG

0005Ø 0000               ORG   Ø        Relocate as required

0007Ø               *LETNUM sorts numbers, symbols, control & letters.
0008Ø               *C = number; V = symbol; Z = control; else, letter.

0011Ø 0000 81 20    LETNUM CMP A  #$20    Space code
0012Ø 0002 25 1D           BCS    STERM   Under $20 is a terminator
0013Ø 0004 81 30           CMP A  #$30    Check for delimiters
0014Ø 0006 25 14           BCS    DELIM   $20-$2F, all delimiters
0015Ø 0008 81 3A           CMP A  #$3A    Now, for numbers
0016Ø 000A 25 14           BCS    EXLET   0-9 recognized (not hex)
0017Ø 000C 81 41           CMP A  #$41    $3A-40, all delimiters
0018Ø 000E 25 0C           BCS    DELIM
0019Ø 0010 81 5B           CMP A  #$5B
0020Ø 0012 25 08           BCS    LET
0021Ø 0014 81 61           CMP A  #$61
0022Ø 0016 25 04           BCS    DELIM
0023Ø 0018 81 7B           CMP A  #$7B
0024Ø 001A 25 03           BCS    LET
0025Ø 001C 4D       DELIM  TST A           Clear Z, C, N
0026Ø 001D 0B              SEV             Flag for delimiter
0027Ø 001E 39              RTS
0028Ø 001F 4D       LET    TST A           Letter. Clear V, C, N & Z
0029Ø 0020 39       EXLET  RTS
0030Ø 0021 36       STERM  PSH A
0031Ø 0022 4F              CLR A           Sets Z; clears all others
0032Ø 0023 32              PUL A           In case it's needed
0033Ø 0024 39              RTS


0036Ø                    END

TOTAL ERRORS 0000
```

Microware Relocatable Assembler And Linker

A Review of the Microware
Relocatable Macro
Assembler and Linker

by E. M. (Bud) Pass, Ph.D.
Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717/4570

GENERAL INFORMATION

RMA is a macro assembler designed to assemble programs written for the Microware OS/9 Level I or Level II operating system for the 6809. Included with RMA is a linker, named RLINK, which allows the construction of an OS/9 module from a group of compatible pre-assembled parts.

RMA is marketed by Microware both to support the new Microware C Compiler and to provide a more powerful facility than the older Microware Interactive Assembler, which is still being marketed.

In addition to its basic ability to recognize the 6809 assembly language and to process symbolic address expressions, RMA has the ability to support conditional assembly, to define macros and invoke them with parameters, and to assemble programs stored in modular form in such a fashion that pieces of a larger module may be separately assembled and linked together. In fact, the assembler (RMA) produces a relocatable object file which must be processed with the linker (RLINK) before OS/9 will allow it to be executed.

The discussion of RMA presented here is intended to familiarize the reader with the concepts of RMA and to introduce the reader to several advanced assembler language and linker concepts, also applicable to many other assemblers. A working knowledge of an assembler such as the Microware Interactive Assembler and an operating system such as Microware OS/9 would be helpful in understanding this article. A brief comparison of RMA with the TSC 6809 Macro Assembler, the Microware 6809 Interactive Assembler, and the OSM 6809 Macro Assembler is also presented.

INVOKING ASSEMBLER AND LINKER

The command lines used to invoke the assembler and linker are as follows:

```
    for the assembler:

    RMA inputfile [-options ...] [>listfile]

    for the linker:

    RLINK [-options ...] mainlinefile [subfile ...] [>listfile]

For example,

    RMA program1 -c -s -x >/p

    end

    RLINK program1 program2 program3 -o=program -n=programm -m -s
```

Only one file is allowed as input to each execution of the assembler. However, this file may employ USE pseudo-opcodes to include additional files in the assembly process. Each execution of the assembler may produce a listing file and a relocatable object file, as controlled by the options described below.

The assembler options are listed below. They are each preceded on the command line by "-" or "--", usually indicating that the option is to be turned on or off, respectively. Each option which may have a value has the default value specified in parentheses below. Up to ten options may be specified.

```
-c (on) suppress listing of conditional assembly lines

-dn (n=66) set number of lines per page for listing to n

-e (on) suppress printing of errors

-f (off) use form feed, not line feed, to advance listing
        to top of form

-g (off) list all code bytes generated

-l (off) list formatted assembler output
```

-o=pathname write relocatable object file to pathname

-s (off) list symbol table at end of assembly

-wn (n=80) set line width to n

-x (on) suppress macro expansions in listing

One mainline file is allowed as input in each execution of the linker. As many sub-files as are required may also be specified in the same linker execution. The mainline file and all sub-files must be relocatable object files produced by the assembler. Each execution of the linker may produce a listing file and an OS/9 module file, as controlled by the options described below.

The linker options are listed below. They are each preceded on the command line by "-".

-bsvpt link C functions to be callable by BASIC09, and provide entry point opt for transfer address

-E=n set edition number to n, rather than 1.

-l=path use path as library file

-Mnnnk| indicate number of pages or kbytes of additional data space to be allocated for OS/9 object module

-m print linkage map providing base addresses in OS/9 object module

-n=name use name as OS/9 object module name

-o=path write OS/9 object module to path with name of last element in path name unless overriden by -n=name parameter

-s print final addresses of symbols in OS/9 object module

-t allow static data to appear in BASIC09 callable module

RMA PROGRAM ORGANIZATION

RMA requires that every program be organized into program sections, to assist it in defining the program and data storage areas required for all OS/9 object modules. These sections are as follows:

PSECT indicates the beginning of the program section, which will contain instructions and constants to be placed into the relocatable object file, and initializes the location counters. Program code generated by a PSECT is always generated relative to zero. There may be only one PSECT section per separately assembled module. The RMB pseudo-opcode is illegal in a PSECT. The PSECT section header may contain a list of parameters, as described below, in which case it is known as a "mainline" module, or it may not, in which case the PSECT name defaults to "program" and all other parameters default to zero. The format of the PSECT section header is as follows:

PSECT name,typelang,attrev,edition,stack,entry

In which the parameters have the following interpretations:

name	PSECT name of up to 20 characters
typelang	OS/9 type/language byte
attrev	OS/9 attribute/revision byte
edition	OS/9 edition number byte
stack	estimate of stack space required by this module
entry	mainline module entry point

VSECT indicates the beginning of a data section, which will contain offsets from the beginning of an area to be established by the linker for each sub-module, as a part of the ultimate OS/9 object module; data areas may be initialized or uninitialized. There may be as many VSECT sections per separately assembled module as required, and the location counters are maintained from one section to the next, but the location counters may not be adjusted. Program instructions may not be coded in a VSECT. The VSECT section header may have one optional parameter, the literal "DP", in which case the direct page location counters are used for this VSECT section, rather then the data location counters normally used for VSECT sections.

CSECT indicates the beginning of a counter module, which may be used as a convenience to the programmer for establishing assembler table values, etc. As many CSECT sections per separately assembled module as required may appear. No code may be produced in a CSECT, so only RMB, EQU, SET, etc. pseudo-opcodes are usually used in a CSECT. The CSECT section header may have one optional parameter, which is assumed to be a value to which to initialize the CSECT location counter; however, if the parameter is not present, the CSECT location counter is initialized to zero.

A program section is normally terminated with the ENDSECT pseudo-opcode. Between the section header (PSECT, VSECT, CSECT) and ENDSECT are normally found assembler-language statements specifying the actions required of the assembler. Certain pseudo-opcodes, such as EQU, SET, NAM, TTL, OPT, PAG, SPC, USE, FAIL, REPT, IFxx, MACRO, etc., may also appear outside of all program sections.

RMA PROGRAM STATEMENTS

RMA accepts standard Motorola assembler free-format source language statements. Each assembler language statement may contain ASCII characters with hex values between $20 and $7e. The end of line indicator must appear after each statement and has hex value $0d (a carriage return). A total line length of 256 characters (including the carriage return) is allowed.

Four optional fields on the source line are recognized by all Motorola-compatible assemblers, including RMA. These consist of the label, operator, operand, and comment fields, separated by one or more space characters, as required. The format just described may be represented as follows:

[label] [operator] [operand] [comment]

In which the label field, if present, must start in the first position, and if not present, must be represented by a space.

If the label field is present and starts with an asterisk, the line is assumed to be a comment line only. Entirely blank lines are also assumed to be comment lines only.

The restrictions and options for each field are as follows:

LABEL FIELD

The label must not be preceded by spaces.

The label must be unique in the local module, unless it is followed by a colon, in which case it must be unique in the entire linked object module.

The label must consist of one to nine of the following characters:

'A' - 'Z', 'a' - 'z', '1' - '9', '$', '.', '_', '@'.

and must begin with a letter.

The label (with optional trailing colon) must be followed by a space.

The label will have a value depending upon the operator on the same line and the program

section in which it is defined.

OPERATOR FIELD

RMA supports all of the standard Motorola 6809 instructions (LDA, STA, LEAX, ...) and essentially all of the standard Motorola 6809 pseudo-opcodes (EQU, SET, FCB, FCC, RMB, PAG, TTL, OPT, ...).

RMA supports several pseudo-opcodes which are specifically oriented towards OS/9 assembler language programs, including the following:

FCS string	which is similar to FCC except it sets the high-order bit on the last byte in string
RZB expression	which is similar to RMB except it clears the included memory to a value of hex zero
OS9 expression	which generates a call to OS/9, and is equivalent to the following code sequence: SWI2 FCB expression

RMA does not directly support the 6800 forms of several of the 6809 instructions (LDAA, LDA A, INX, ...), although it allows macros to be written to support them, as described below.

OPERAND FIELD

Operands are considered as composed of one or more expressions, separated by commas.

They are usually evaluated for the expression value and addressing mode specified.

In case an instruction requires no operand, the operand field is considered part of the comment field.

COMMENT FIELD

This field is optional and is composed of the remainder of the characters on the line after the operator and operand fields.

The only processing performed on this field by RMA is to place it on the listing file.

EXPRESSIONS

Expressions consist of combinations of symbols, constants, and program/data/offset counters separated by operators. Arithmetic is done in 16 bit integer precision, such that all values must be integers between -32768 and +32767 for signed computation or 0 to +65535 for unsigned computation. If a signed value must be truncated to eight bits, the value must lie between -128 and +127, and if an unsigned value must be truncated to eight bits, the value must lie between 0 and 255; otherwise an error message is generated.

Expressions must not contain spaces, and the expression is terminated when a space, carriage return, or unexpected character is found. They are evaluated in the operator precedence order shown below, with left-to-right order for operators with the same precedence.

Following is a summary of the formats for constants:

base	prefix	characters
Decimal	none	0-9
Binary	%	0-1
Octal	@	0-7
Hexadecimal	$	0-9,A-F,a-f
ASCII	'	$20-$7e

Following is a summary of the operators, in order of decreasing precedence:

operation	operator	format
grouping	( )	(value)
negation	-	- value
not	^	^ value
and	&	value & value
or	!	value ! value
multiply	*	value * value
divide	/	value / value
add	+	value + value
subtract	-	value - value

Symbols are labels, as described above. The current program counter value is represented (in a PSECT) by the character '*'. The current data counter value is represented (in a VSECT) by the character '.'.

RMA MACRO DEFINITION AND INVOCATION

Many assemblers, including RMA, possess the ability to process macro definitions and macro invocations, although the details of the implementations vary widely, even among Motorola-compatible assemblers.

Macro capabilities can assist the programmer in structuring assembler code and can save much coding and maintenance time. Standardized macro libraries are used in many programming systems to provide a very-high-level command capability in assembler language.

Macros differ from subroutine calls in that the source language lines represented by the macro are inserted into the program, perhaps with individual custom changes to the code, every time the macro is called, rather than being placed at one location within the program and being branched to for each call.

The first line of an RMA macro definition must have an operator field containing MACRO and a nonblank label field, which is used as the name of the macro being defined. The last line of the macro definition must have an operator field containing ENDM, but the label field is ignored. The macro text between the MACRO and ENDM pseudo-opcodes is saved in memory and on disk for later reference.

To invoke the text referenced by a macro name, the macro name is used as if it were an operator. In fact, the standard 6809 instructions and the RMA pseudo-opcodes may be redefined, as macro names override operator names in RMA. If any parameters are to be passed to the macro, then the macro name is followed by a space and one or more expressions, separated by commas, in the operand field.

While a macro is being expanded, parameters are substituted symbolically wherever they appear. Parameters are represented within the macro text by the characters '\n', where n is a digit from 1 to 9. If an parameter is referenced, but is not defined on the line from which the macro is called, it is assumed to have a null value. Parameter values may not be changed within a macro.

Two special macro parameters are supported by RMA to assist in the use of macros. They are as follows:

\ ln returns the length of parameter n in bytes.

\ 0 returns the number of parameters in current invocation.

Macros may invoke other macros, or even invoke themselves recursively. Parameters are referenced locally in each macro. If a macro is called from within another macro, the current macro's parameters are pushed on to a stack for later retrieval, in a

fashion analogous to the saving and restoring of machine registers during a subroutine calling process. The macro calling line may use the parameter substitution process to pass a parameter to a subsequent macro.

An example of the use of macros appears below. It extends the RMA language to handle many 6800 and 6801 compatible mnemonics. Note that there may be no space in such 6800/1-style mnemonics as "LDA A", and that character case is significant.

```
CLC     MACRO
        ANDCC #$FE
        ENDM
CLI     MACRO
        ANDCC #$EF
        ENDM
CLV     MACRO
        ANDCC #$FD
        ENDM
CLZ     MACRO
        ANDCC #$FB
        ENDM
CPX     MACRO
        CMPX @1
        ENDM
DES     MACRO
        LEAS -$01,S
        ENDM
DEX     MACRO
        LEAX -$01,X
        ENDM
INS     MACRO
        LEAS $01,S
        ENDM
INX     MACRO
        LEAX $01,X
        ENDM
LDAA    MACRO
        LDA @1
        ENDM
LDAB    MACRO
        LDB @1
        ENDM
LSRD    MACRO
        LSRA
        RORB
        ENDM
PSHA    MACRO
        PSHS A
        ENDM
PSHB    MACRO
        PSHS B
        ENDM
PSHX    MACRO
        PSHS X
        ENDM
PULA    MACRO
        PULS A
        ENDM
PULB    MACRO
        PULS B
        ENDM
PULX    MACRO
        PULS X
        ENDM
SBA     MACRO
        PSHS B
        SUBA ,S+
        ENDM
SEC     MACRO
        ORCC #$01
        ENDM
SEI     MACRO
        ORCC #$10
        ENDM
SEV     MACRO
        ORCC #$02
        ENDM
SEZ     MACRO
        ORCC #$04
        ENDM
STAA    MACRO
        STA @1
        ENDM
STAB    MACRO
        STB @1
        ENDM
TAB     MACRO
        TFR A,B
        TSTA
        ENDM
TAP     MACRO
        TFR A,CC
        ENDM
TBA     MACRO
        TFR B,A
        TSTA
        ENDM
TPA     MACRO
        TFR CC,A
        ENDM
TSX     MACRO
        TFR S,X
        ENDM
TXS     MACRO
        TFR X,S
        ENDM
WAI     MACRO
        CWAI #$FF
        ENDM
```

AUTOMATIC MACRO LABEL GENERATION

RMA is capable of automatically generating labels in macros of the forms '@@xxxyyyyy' or 'yyyyy@@xxx', in which 'xxx' represents a decimal number and 'yyyyy' represents zero to five characters added for uniqueness. An automatically-generated label is represented symbolically by '@@', followed by zero to five characters. Each time a macro with automatic labels is called, a counter is incremented, such that every call will generate a new set of labels.

An example of automatic label generation is the macro defined below.

```
STRING  MACRO
        LEAX @@A,PCR
        LBSR PSTRNG
        BRA @@B
@@A     FCS "@1"
@@B     EQU *
        ENDM
```

It could be called as follows:

```
        STRING POINT1
        :
        :
        STRING POINT2
```

The code generated (on the first call) would be as follows:

```
        LEAX @001A,PCR
        LBSR PSTRNG
        BRA @001B
@001A   FCS "POINT1"
@001B   EQU *
```

The code generated (on the second call) would be as follows:

```
        LEAX @002A,PCR
        LBSR PSTRNG
        BRA @002B
@002A   FCS "POINT2"
@002B   EQU *
```

CONDITIONAL ASSEMBLY USING RMA

The use of conditional assembly pseudo-opcodes can assist in structuring and generalizing programs written using RMA. A very convenient use of the RMA

conditional assembly capabilities is for customizing programs to a set of parameters, since the parameters may be used to control the assembly. An excellent example of such use is to define a symbol representing whether the program is to be used with OS/9 Level I or Level II, and to include the appropriate code in either case.

The syntax of the IFxx pseudo-opcode is as follows:

```
IFxx expression
:
:
[ELSE]
:
:
ENDC
```

IFxx will cause the lines up to the next matching ENDC or ELSE to be included only if the expression has a desired relationship to zero. The 'xx' may have one of the following values:

EQ,NE,LT,LE,GT,GE

indicating the desired comparison to zero (equal, not equal, ...). The expression may not contain external or undefined symbols.

If a matching ELSE is found before the matching ENDC, the test condition is reversed for the succeeding lines.

The syntax of the IFP1 pseudo-opcode is as follows:

```
IFP1
:
:
[ELSE]
:
:
ENDC
```

IFP1 allows for assembly of the included lines only during pass 1 of the assembly process. This may be used for including large symbol definition files without requiring that RMA read them during pass 2, after it has already read them during pass 1, thus speeding up the assembly process.

The syntax of the REPT pseudo-opcode is as follows:

```
REPT expression
:
:
ENDR
```

REPT repeats the included lines the number of times specified by the expression on the REPT pseudo-opcode. REPT pseudo-opcodes may not be nested. The expression may not contain external or undefined symbols.

The syntax of the FAIL pseudo-opcode is as follows:

```
FAIL text-string
```

FAIL prints its operand as an RMA error message and increments the error counter. It is typically used with conditional assembly and in macros to indicate illegal conditions in the source code.

Unlike high-level language statements, the RMA conditional assembly pseudo-opcodes actually control the inclusion, exclusion, or repetition of the source language statements in their range, rather than controlling the execution of the statements This implies that the RMA conditional assembly pseudo-opcodes are evaluated at assembly, not execution time.

MODULAR RMA SOURCE LANGUAGE

The syntax of the USE pseudo-opcode is as follows:

```
USE filename
```

The USE pseudo-opcode may be used to introduce additional source language files into a file being assembled.

It is especially useful for large programs and for source libraries, and provides some of the capabilities at the source language level normally provided by a linker at the object language level It assists in the development of structured, modular assembly source language programs. USE pseudo-opcodes may be nested to a limited extent.

RMA LINK-EDITOR

Included with RMA is RLINK, a link-editor. The combination of RMA and RLINK allows assembler language programs to be separately assembled and then linked together to form larger OS/9 object modules.

RMA does not generate executable an OS/9 object module directly, but instead generates a relocatable object file containing a special representation of the program, which also includes information concerning external references and global labels in the program.

RMA generates each relocatable object file relative to $0000 in both program and data location counters. The PSECT and VSECT section headers and trailers assist RMA in determining the program and data space sizes for each program. Mainline programs (those containing complete PSECT header information) provide additional information for RLINK to enable it to correctly generate the final OS/9 object module.

RLINK accepts a list of relocatable object files, starting with the mainline file, concatenates the program code segments, resolves the external and global references, and resolves the references to the data and direct-page areas throughout the relocatable object files. If there are no unresolved external references and all relocatable object files are properly formatted, RLINK is able to construct the resultant OS/9 object module.

Through proper planning, the RMA and RLINK facility of separately assembled modules, plus the USE pseudo-opcode facility of modular source modules, may be combined to form a powerful development tool for the system designer building large and versatile systems.

COMPARISON WITH OTHER ASSEMBLERS

RMA has many features which are similar to the OSM (Lloyd I/O Assembler) and MIA (Microware Interactive Assembler) assemblers, and others which are different from either of them. The primary similarities among all the assemblers are related to the use of the Motorola-compatible format and Motorola 6809 mnemonics. Often, when a product (such as an assembler) does not possess one feature, it has another feature to compensate for the missing feature.

RMA and MIA do not support the 6800-compatible instruction formats (such as LDAA and INX), the expanded FCC statement, the IF, IFN, IFC, IFNC, ENDIF, ERR, DUP, ENDD, RPT, and EXITM pseudo-opcodes, the shift operators '<<' and '>>',

the relational operators '=', '<', '>', '<=', '>=' and '<>', command line parameters, and the BEC, BES, CLF, CLZ, LBEC, LBES, SEF, and SEZ instructions, as does the OSM assembler.

RMA, MIA, and OSM support the logical 'or' and 'not' operators in symbolic expressions, but RMA and MIA use '!' and '©', whereas OSM uses '¶' and '|' to represent the same operations. Similarly, RMA uses the REPT and ENDR pseudo-opcodes to specify source language repetition, whereas OSM uses DUP and ENDD for the same function.

Only MIA supports the MOD and EMOD pseudo-opcodes and an interactive mode of operation, but MIA does not support the RZB pseudo-opcode supported by both RMA and OSM.

Only RMA supports the PSECT, VSECT, CSECT, ENDSECT, and certain other pseudo-opcodes oriented toward the generation of relocatable object files which may be linked together. RMA does not support the MOD, EMOD, and SETDP pseudo-opcodes because of conflicts with the linking process.

RMA and RLINK cannot produce a file of object code without a standard OS/9 header. This significantly reduces its ability to produce non-OS/9 object code, such as its potential use with macro libraries to generate code for another type of computer system. This is in addition to the OS/9 problems of always requiring that all OS/9 object modules start at logical address $0000 and be contiguous.

The lack of OSM's (and TSC's) IFC and IFNC pseudo-opcodes further restricts RMA from performing cross-assembly for other systems, since it is then not possible to determine the difference between 'X' and '(X)', which may be quite significant in some assembler languages. IFC and IFNC differ from the RMA IFxx statements in allowing the literal inspection of a parameter, rather than the testing of the value of an expression, possibly involving a parameter.

The OPT pseudo-opcode operands are different among all the assemblers.

CONCLUSIONS

RMA does not have the field experience enjoyed by the MIA and OSM assemblers, simply because it is so new. This does not imply that RMA has more bugs or problems than any of the other OS/9 assemblers. All assemblers have known (and probably unknown) bugs, inconsistencies, and peculiarities. It does imply that RMA does not yet have the name recognition of the other assemblers, a problem which it will overcome, given time.

A known bug in an early version of RMA (which has been fixed by Microware) concerns the use of the '/' operator in symbolic expressions. For example, in the following statement:

```
label equ $8382/256
```

RMA would compute the value of 'label' as $FFB4, not as $0083, which implies that the division returned a signed, not unsigned result. This would have negated the use of division by a power of 2 to simulate a bit-shift operation, as RMA has no shift operators.

This article has discussed the Microware Relocatable Macro Assembler for OS/9 Level I and Level II for the 6809-based systems. It presented many of the features supported by RMA that differ from the features supported by other OS/9 assemblers, including the provision for a link-editor to enable the separate assembly of modules and the combination of them into larger OS/9 modules.

Note: All occurances of the following shown symbols are to be transposed to the actual symbol. These changes are due to typewheel considerations on our computer.

ACTUAL	SHOWN
Up Arrow	© (Copyright)
L Brace	§
R Brace	†
Backslash	®

68K Experiences

Hello!

In the april issue of 68' Micro Journal, you were asking your readers to write articels about 68000 technology. Concerning to this request, I have written an article about "68K EXPIRIENCES" which you may publish in 68' Micro Journal. I have send you a paper-copy of the article and a 5" single sided, single density, 40 tracks disks with the article written on it in TSC EDITOR FORMAT. The filename is "MICRO68.TXT". I hope that you are able to read this disk. To give you an idea about who I am: I am working on 68' systems since about 5 years (my age is 20). My first 68' system was a 6802 based system, rather small but good for self- education. After about 2 years I changed to a larger 6809 system and since 1983 I am also working on a 68000 system which is described in the articel. About three years ago I started to do professional work for a number of companies and organizations of different industries: computer-, building-, pharma-, aeronautic and spaceflight- and military-industries. Besides writing the programs I was also developing the hardwere, if there was no product on the market that was fitting to the application. The programs I have written are mainly real-time programs. At time I am doing my duty military service at the "FEDERAL GERMAN ARMED FORCES UNIVERSITY" in Hamburg. Here I am developing a system that is processing usable data from measured values it receives from a tank-test-system. End of september this year I will finish my military service. Then I want to study computer technology at university. However, I would prefer to study in the U.S. rather than in West-Germany. So I am searching for a possibility to work for a U.S. company or organization in order to earn the money needed to pay studling at a U.S. university. I would be pleased if you could remark this in 68' Micro Journal, of course only if it's possible and not against editors rules.

Would you please send me a copy of that issue of 68' Micro Journal in which my articel is published because I am getting your magazine only irregulary via a library.

Best wishes for your and 68' Micro Journals future.

Yours'

sender:
Marco Salzwedel
Wiesenring 59
D-6093 Floersheim/4
West Germany

Tel.: phone code for West-Germany + 40/6510153

or 6145/32772

68K Experiences

In the april issue of 68' Micro Journal the readers were asked to write some articles about their experiences with 68000 systems and the 68000 itself. I want to use this offer to write a bit about 68000 activities in West-Germany. This article might especially be interesting for people who where using up to now an eight-bit system but are thinking about to change to a sixteen bit system.

Last year a german company that is active in the aeronautic and spaceflight market asked me to simulate an eight channel oscilloscope on computer basis. This oscilloscope was needed for medical use. Therefore it should offer some analysis comfort and an analog-value measuring frequency of 1000 Herz per channel. It was obvious that this task could hardly be realized with an eight-bit system. So I was forced to use a system that is offering high speed and a lot of memory in order to solve this task. I choosed the 68K system from ELTEC elektronik in Mainz / West-Germany.

A 68000 SYSTEM WITH A NEW CONCEPT

The 68K system consists of single-boards in double Europe format (the size of a double Europe format board is about 160 mm * 234 mm). The boards are connected via a VME bus, a bus that meets the requirements of 16- and 32-bit microprocessors. This modularity enables the user to get a system that fits to his individual ideas. One main board of the 68K system is the EUROCOM 3 computer. This board offers a 68000 or 68010 CPU with a cycle rate of up to 12 Mhz. 256 KByte or 1 MByte of RAM and 64 KByte of EPROM are available on this board. In addition there is a floppy disc controller for 5" and 8" drives, two serial interfaces for different synchronous and asynchronous modes with data transfer rates of up to 1 Mbit per second and a CMOS realtime clock with battery buffering. The 68K systems makes good use of the special capabilities of the 68000 family and the VME bus. Therefore it is possible to realize a multiprocessor system which pushs the frontiers of your systems computer- power far away. For instance if you have to do a lot of calculation work (i.e. Fourier transformations) in realtime, you could add a SAC (Stand Alone Computer) board to your system. The SAC is a board with a 68000 microprocessor and 64 KBytes of fast static RAM (even at a clock frequency of 16 Mhz no wait cycles occur). With the help of the on board floating-point unit, i.e. a 64 bit floating point division may be executed within 20 microseconds including the data shift from the CPU register to the floating-point unit and back if a clock frequency of 12 Mhz for the CPU and 8 Mhz for the FPU is used. Another important time saving factor is the use of interrupts. The interrupt capabilities of the 68000 are fully implemented in the 68K system, with all seven interrupt levels. Other boards in the series are the GRAZ for fast high resolution (512 * 512 points, 7 colors) graphic, a 12 bit A/D D/A converter board with 8 A/D and 4 D/A channels and conversion times of about 8 microseconds, a parallel I/O board and RAM extension boards with a capacity of 512 MBytes or 2 MBytes. Up to now the CP/M-68K and the OS-9 operating system are available. You may get Pascal, C and Fortran compilers as well as BASIC interpreters, Assemblers and Editors. Additionally there exists software which supports the special hardware of the 68K boards.

USING THE 68000

I don't think that it is a good idea to use a 68000 CPU in a computer that is build to solve tasks which easily could be done by an eight bit machine. In this case the more complex structure caused by the external 16 bit Databus and 23 bit Addressbus would mean nothing more than a waste of computerpower or money. The 68000 is designed to solve complex and data intensive problems within a short time. Therefore real-time systems like driving machines, simulations or processing data from measured values are applications where the 68000 is often a good choice. But on the other hand, just using a 68000 doesn't make a powerful computer. It's a well known statement, that the weakest part of a system defines the powerlimits of the system. Does it really make big sense to build a 68000 computer with some dozen KBytes of RAM/ROM while 16 MByte of direct addressing range are available, with slow memory and peripheral interface circuits while the CPU may be driven with a clock frequency of up to 16 Mhz, without the use of the interrupt capabilities of 255 interrupt vectors and 7 interrupt priority levels or with a concept that doesn't make use of the multiprocessorsystem support of the 68000? It's like a Porsche motor won't make a racing car out of a tractor. Of course it's not always necessary or possible to use all the 68000 capabilities in a new computer, but designers should have a concept in mind that fits to the new structure that was introduced with the 68000. Compatibility to former standards should have only secondary importance. I think the 68K system is an example of a lucky realization of these ideas.

However, not only the hardware-, also the software-structure must fit to the new system structure in order to realize efficient programs. When I first time developed a timecritical routine in 68000 Assembler I was very impressed about the powerful commands but I still had the limits of my 6809 system in mind. So I designad a routine that was using rather few memory for data storing but therefore had to do a lot of calculations more than once. The result was an impressive compact routine which was also impressive slow. The best possibility to speed up the routine was to NOT do a lot of calculations more than once, but instead store more results for future use. In this special case this meant that I needed 64 KBytes RAM for data storing instead of 16 KBytes. But does this make a difference if you may use a capacity of 1 MByte that too might be expanded? However, this new routine was much faster and this was exactly what I wanted. Now please don't mix up code density and data density. High code density (that is as few instructions as possible executed) still means high speed. But high code density often implements low data density (the same data is storad in different formats i.e. the data is stored in the 'original' format AND in the format you get after a calculation). Because the 68000 offers a large direct addressing space, it is no more impossible to speed up a routine by the help of low data density. The machine cycles needed to execute a 68000 command differ very much. i.e. to move a long word (32 bit) from one CPU register to another CPU register will take four machine cycles while moving a long word from one memory location to another one needs 29 machine cycles. A signed 16-bit division may use up to 158 machine cycles. This makes it necessary to carefully choose the commands especially in routines that are executed a lot of times within a short period of time. In real-time systems often different actions have to be repeated with an exact frequency (i.e. the 'oscilloscope' I was developing has to measure the incoming analog data 1000 time per seconds all the same how much data processing work has to be done under the individual circumstances). This is best realized with the help

of timers that drive interrupts. Because the 68000 offers a lot of interrupt vectors it is no more necessary to step through a long list in order to recognize which device was requesting an interrupt. Instead the interrupt vector may point directly to the start of the individual interrupt service routine. This is a great time saving especially when there are a lot of interrupting devices and short interrupt service routines. Additionally the seven interrupt priority levels enable a higher-priority device to interrupt the processor during execution of an interrupt routine. When used carefully this priority orientated interrupt structure ends in a self-optimization of the program execution: important works are indicated with a high priority interrupt and vice versa. Now the processor will do less important works (like updating a relatively uninteresting result on screen) only if he has to do 'nothing better' at time. I realized this concept in my 'oscilloscope' program and it's doing real good work. However, although when using all these features of the 68000 there is still a limit because the processor can do only a certain amount of work within a limited period of time. To overcome this two or more processors have to share the work. Until now I didn't have the chance to experiment with a multiprocessorsystem, so I don't know how to realize it best. However, I am going to do some research on that in the near future. One idea is to use one processor as the 'master processor' that knows which works have to be done next. This 'master processor' should handle new tasks over to the 'slave processors' whenever they have finished a work. If this would be possible you could speed up a system just by connecting another processor board to it. This concept would be the best because the computer power is used most efficiently and the software must not be changed when increasing the number of processors. The other extreme is that every 'slave processor' will do only it's strictly predefined work while the 'master processor' is just asking the 'slave' to do this work when it's needed. This concept is much less efficient but it should be no problem to realize it. I guess it's possible to develop a concept anywhere in between these two ideas.

THE EFFICIENCY OF MIDDLE AND HIGH LEVEL LANGUAGES ON THE 68000

If you want to write powerful Assemblerprograms for the 68000, you have to take use of the special features of the 68000. When programming in a middle or high level language the same is valid for the compiler or interpreter used: A compiler or interpreter that was designed to run on a 6809 machine and that was just adapted to run on the 68000 won't produce the best programs. Besides this also the 'intelligence' of the code translating program is an important aspect. Is the translator able to recognize individual situations where he could save some commands or is he always translating in the same way? To test this I have written a small 'senseless' routine, the flowchart is shown in picture 1. Listing 1 is the 'C' version of the program, listing 2 the assemblercode produced by the 'C' compiler and listing 3 is the 'handwritten' assemblerversion of the routine. The 'C' routine was translated by the 'C' compiler from Digital Research that is running under CP/M-68K. The result is rather interesting. If you have a look at listing 2 you see that the compiler moves the variables at the beginning of a 'C' line from memory to the CPU register and after execution of the 'C' line back to memory. And he does it also when the same variable is needed in the next line again. This result indicates that the 'intelligence' of the 'C' compiler ends at the end of the 'C' line (or better say 'C' statement) it is translating. However, the 'C' compiler shows a certain 'intelligence' within a line: the 'b = a * 2' multiplication is realized via a one-position-shift-left command and not via the time consuming 'imul' routine from his library that it is using in the 'c = a * 3' statement. The comparison of the program execution times ends in a bad result for the 'C' routine: 92 seconds against 11 seconds which the 'handwritten' assemblerroutine is running. But this relation is not representative at all. If you take away the 'c = a * 3' statement in both routines the relation changes to 22:8 seconds. Although this relation is not too bad it is still obvious that programming in a middle or high level language takes quit a bit of the 68000 power away. Therefore it is often not possible to use one of these interpreters or compilers for writing a timecritical routine. On the other hand programs normally do not consist of timecritical routines only. Often there are also complex routines in the program where the timefactor is of secondary importance. Just think about a user-computer dialog routine. Few peoples are willing to write these complex routines in Assembler. The way out of this dilemma is not out of sight: middle and high level language translators normally support the use of assemblerroutines as subroutines. This feature enables the user to write the slow and clumsy dialogroutines etc. and the small main-routine in an easy to write and maintain higher level language while the high-speed routines may be written directly in assembler. I am using this 'language mix coctail' concept since a long time with great success. Especially when using the 68000 which is predestinated for real-time applications this concept seems to demonstrate how assemblers and higher level languages are used best.

Although the 68XX and 68XXX family processors are not as famous in Europe as they are in the U.S., I hope that I was able to point out that there are still good 68' products on the european market and also people that like to use these processors.

```
main()

{
long a,b,c;

while(a < 10000000)
  {
  a = a + 1;
  b = a * 2;
  c = a * 3;
  }
}
```

Listing 2: This assembler code is produced by the 'C' compiler

```
.globl main
.text
_main:
~~main:
~a=-4
~b=-8

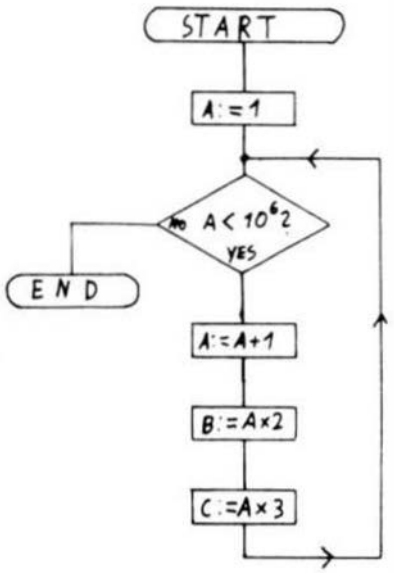
  link R14,#-12
  move.l #1,-4(R14)        a = 1
L3:
  move.l -4(R14),R0
  cmp.l  #10000000,R0      a < 1000000 ?
  bge    L2                branch if no
  move.l -4(R14),R0                      move 'a' from memory to CPU register
  add.l  #1,R0             a = a + 1     do the calculation
  move.l R0,-4(R14)                      move 'a' from CPU register to memory
  move.l -4(R14),R0                      move 'a' from memory to CPU register
  asl.l  #1,R0             b = a * 2   times two multiplication realized by one
  move.l R0,-8(R14)                    position shift left
  move.l R3,-(sp)          c = a * 3
  move.l -4(A6),-(sp)
```

```
  jsr   lmul
  add   #8,sp
  move.l R0,-12(R14)
  bra   L3
L2:
  unlk  R14
  rts
.data
```

Listing 3: The 'handwritten' assemblerroutine. Execution time: 11 seconds

```
      MOVE.L #1,D0
COMCOU CMP.L  #10000000,D0  a < 1000000 ?
      BEQ   ENDCOU          branch if no
      ADDQ.L #1,D0          a = a + 1
      MOVE.L D0,D1
      ASL.L  #1,D1          b = a * 2
      MOVE.L D1,D2          c = a * 3
      ADD.L  D0,D2
      BRA    COMCOU
ENDCOU RTS
```

PICTURE 1: PROGRAM 'SENSELESS'



Cyclic Redundancy Check

Specialized CRC hardware improves the performance of OS-9™

By: Mike Magnus

One of the unique features of Microware's OS-9 operating system is its use of a Cyclic Redundancy Check (CRC) to verify data integrity. A CRC is a commonly used error detection technique, usually associated with mass-storage systems and serial data transmission. The use of this technique greatly enhances reliability by detecting errors of various types before they can have adverse effects on the system. However, as with all good things, there is a price to be paid. In this case, the price is speed. The use of the CRC adds a certain amount of system overhead which reduces overall system performance. While this overhead cannot be eliminated completely, it can be reduced significantly through the use of specialized hardware.

OS-9 uses the concept of "Memory Modules" to manage items that are loaded into the system memory. A Memory Module may be an executable 6809 object code program or subroutine, a data block, or a program in an intermediate code created by a high-level language. In addition to the actual data, each module includes a header to describe its contents and a three byte CRC value. When OS-9 loads a module into memory from a mass-storage device, or when it locates a module in memory during its initialization process, a CRC value is computed from the data in memory. If the computed value does not agree with the value appended to the module, an error is reported and the module can not be used. This process verifies that the data in memory matches the original contents of the module; checking both the loading process and the memory in which the module is loaded.

OS-9 normally computes the CRC by using a software algorithm. When a module is created, each byte of the module is successively passed through the algorithm, resulting in a 24-bit value based on the new byte and the previous CRC value. (Before the first byte of a module is processed, the CRC is initialized to all ones.) The CRC is accumulated in this way until the whole module has been processed. The final 24-bit CRC is then complemented and appended to the module. When a module is loaded into memory, the same algorithm is used to verify its contents.

The CRC computation is a relatively slow process that requires considerable CPU time. It accounts for a major portion of the time required to load a module from disk into memory and make it available to the system. Little can be done to improve the performance of the algorithm used to compute the CRC. The algorithm is straightforward, with no room for improvement. However, by using high-speed hardware to compute the CRC, a considerable increase in speed can be realized by replacing of the CRC software in OS-9 with a hardware equivalent.

The GIMIX™ CRC GENERATOR BOARD implements the OS-9 CRC generation algorithm in hardware. When interfaced to the operating system kernel its operation is completely transparent to the user. The board does all CRC computations performed by OS-9 or programs that use OS-9's CRC related system calls (F$CRC and F$VModul). Since most, if not all of the programs that create Memory Modules (assemblers, compilers, etc.) compute the CRC value by calling OS-9's internal CRC algorithm, an improvement in assemble/compile times can also be realized.

The software algorithm requires a minimum of 111 machine cycles (55.5 microseconds in a 2 MHz system) to process 1 byte, or approximately 455 milliseconds for an 8K module. In contrast, by using the memory-to-memory DMA on the GMX™ 6809 CPU III to transfer the data to the CRC GENERATOR BOARD, CRC values can be computed and accumulated at 1 byte per microsecond in a 2 MHz system, or approximately 8.2 milliseconds for 8K bytes. The software method also requires additional time to fetch each byte from memory and pass it to the CRC algorithm as a subroutine call. This additional overhead nearly doubles the time required to process each byte by the software method. The only overhead added by using the hardware method is the few microseconds needed to set up the DMA registers on the CPU III

(once for the entire module) and read back the final value from the CRC board.

The following timing comparison was made, using the same 2 MHz system, with and without the CRC board. In each test an OS-9 procedure file was used to "load" Basic09™ into memory 100 times. The procedure file consisted of the command "date t" followed by "load basic09" 100 times, and finally "date t" again. The time required to run the procedure was divided by 100 to obtain the load times shown. NOTE: In each case the "date" and "load" commands were already in memory when the test procedure was run. The times include the overhead required to process the procedure file.

Time Required To Load Basic09

With software CRC:	With hardware CRC:
3.84 Seconds	1.09 Seconds

Load time is affected by the position of the file being loaded within the directory structure of the disk, the type of disk, and the amount of other activity on the system. The above comparison was made using a 47 Mbyte hard disk, with Basic09 located near the beginning of the directory, and no other activity on the system.

To a single user, loading a single program, a few seconds may not make much difference. But, consider a multi-user system with many users, where new modules are loaded frequently. Each time a module loads, it "steals" time that could be used for real work. Or consider a typical "C" compilation which must load several different modules in the process of compiling a program, and may also need to generate a CRC for the output module. Even a small improvement in individual load times can add up to a considerable improvement in overall system performance.

The hardware CRC generation method suffers from one minor disadvantage. Unlike the software algorithm used by OS-9, the hardware CRC generator is non-sharable. Only one process at a time may use the board. Once a CRC computation is begun, be it a single byte or an entire module, it must be completed without interruption. In order to prevent multiple processes from accessing the hardware simultaneously, system interrupts must be masked during a CRC computation. The length of time that the interrupts remain masked depends on the size of the module being verified. (Module producing programs like compilers and assemblers compute the CRC on a byte-by-byte basis; interrupts are only masked for a short time as each individual byte is processed.)

In the majority of applications this will not pose any problems, even when loading very large modules. However, some delay in interrupt response is possible if an interrupt occurs during validation of a large module. In certain special applications where real-time interrupt response is extremely critical, especially if the interrupting device is a standard (non-intelligent) I/O board or device, it may be preferable to use the software CRC rather than the CRC board, or to keep modules in memory at all times so that they do not need to be loaded more than once.

The GIMIX CRC Generator Board

The GIMIX CRC Generator Board is a hardware implementation of the CRC algorithm used by OS-9. It uses a proprietary circuit, designed for GIMIX by Terry F. Ritter, and occupies one 30-pin I/O slot in an SS-50 bus system. The board is currently available only for GIMIX systems using the GMX 6809 CPU III and the GMX III Support ROM version of OS-9.

Included with the board is a software update with two versions of the OS-9 kernel (OS9p1); one configured for I/O slot #6 and one for slot #7. The board may be used in either of these two slots, by selecting the appropriate version of OS9p1. Existing systems are easily updated by installing the board and the appropriate software. No other changes to the system are required and the system can be returned to normal (software CRC) operation by changing boot disks.

In the Support ROM version of OS-9 GMX III, the PROM-resident portion of OS-9 (OS9p1) is moved to disk and replaced by a menu-driven "monitor" program. The monitor includes 3 separate memory tests that can be used anywhere in the 1 Mbyte address space, and bootstrap routines for loading OS-9 from either floppy or hard disk. The Support ROM is available in 2 versions. The first for OS-9 only systems, includes two different Support ROM monitors; one menu driven and one that boots OS-9 directly on power-up. The second version of the Support ROM includes the menu-driven monitor for OS-9, and the GMXBUG-09 monitor for use with GIMIX FLEX™. The OS-9/GMXBUG version of the Support ROM package includes the GIMIX FLEX O.S. and GIMIX VDISK™, a virtual disk (RAM disk) program for FLEX.

The price of the GIMIX CRC Generator Board, with software update, is $275.00. For systems that do not already have the Support ROM version of OS-9 GMX III; the board, with an update to the Support ROM version, costs $400.00 for the OS-9 only version or $575.00 for the OS-9/GMXBUG-09 version. The OS-9/GMXBUG-09 version includes GMX FLEX and VDISK, a virtual disk (RAM disk) program for FLEX. Orders must include the original OS-9 system disk which will be updated and returned.

For ordering information contact:

GIMIX, Inc.
1337 W. 37th Place
Chicago, IL 60609
(312) 927-5510 * TWX 910-221-4055

GIMIX and GMX are trademarks of GIMIX, Inc.
OS-9 & Basic09 are trademarks of Motorola and Microware Systems Corp.
FLEX is a trademark of Technical Systems Consultants
VDISK is a trademark of James L. Arbuckle

Music On The 68000

CLELL A. DILDY, JR.
2401 W 27th St.
Panama City, FL 32405

Don Williams
68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Dear Don

Having recently acquired a 68000 Educational Computer Board (MEX68KECB/D2) from Motorola, I decided to try my hand at programming the 68000. The program of choice of course was a music program that would play "Country Roads". Which would mean that I could use the same set of data that I had used to play the song on the COCO.

The program is organized much like the original 6809 program. The registers were utilized as much as possible. D0 is used as an accumulator. D7 is used as a counter. D1, D2, D3 and D4 were used as address offsets. A0 is used much like the X index register. A5 and A6 were used as fixed pointers to the two

tables. The data for the music was disassembled with "Dynacite", manipulated with the editor, and then appended to the source.

The program was assembled using TSC's 68000 cross assembler. A listing of the Binary file is also enclosed. You will notice that it is in the Motorola S1-S9 hex format. The Binary file was down loaded from my SS-50 system to the single board computer at 9600 baud.

It was of course necessary to provide an audio output from the single board system. A D/A converter was made using an Allen Bradley R-2R network and a two pole filter. The enclosed schematic shows how this was done. Next the data had to be sent from the computer to the network. The printer port was chosen for this as it had eight bit output. Since the 68230 was new to me, it took a while to figure that the handshakes had to be turned off or the chip would only accept one piece of data and then hang up.

Since the play loop is much simpler for the 68000 than for the 6809, it came as no suprise that the time through the 1000 was shorter. The ratio between the two devices is about 1.8 to 1. I had expected this number to be much greater. But the real surprise came when I looked at the number of bytes used and found that the 68000 actually took more bytes of memory than the 6809 to do the same job. After writing this program I see where the 6809 program could be written a little shorter and faster. This would close the gap a little bit more. The only conclusion that I can draw is that the 68000 is not all that much greater than the 6809. In fact a good assembly language program on the 6809 should run rings around a high level language program on the 68000.

I hope this will be of some interest to you and your readers. I would be interested in any other experience with the 68000 vs. the 6809 as one brick doesn't make a load.

Clell Dildy

Clell Dildy

```
*TAB 9,20,40
*
        SPC     20
*****************************************
*                                       *
* PROGRAM TO PLAY FOUR PART MUSIC ON    *
*      THE M68000  MICROCOMPUTER        *
*                BY                     *
*         CLELL  A.  DILDY  JR.         *
*          DECEMBER  17, 1983           *
*                                       *
*         USING AN ALGORITHM BY         *
*             HAL CHAMBERLIN            *
* PUBLISHED IN THE SEPT 1977 ISSUE OF   *
*                 'BYTE'                *
*                                       *
*****************************************
        PAG
        NAM     MUSIC68000
        OPT     PAG
*
*
TEMPO   EQU     $C0
PORT    EQU     $10000
TUTOR   EQU     228
*
*
        ORG     $1000
*
*
* WAVEFORM TABLE FOR ORGAN SOUND
*
WFT     DC.B    $0D,$0E,$0F,$11,$12,$14,$15,$16
        DC.B    $18,$19,$1B,$1C,$1D,$1F,$20,$22
        DC.B    $23,$26,$27,$29,$2A,$2D,$2D,$2E
        DC.B    $30,$31,$32,$34,$35,$37,$38,$38
        DC.B    $39,$3B,$3B,$3C,$3C,$3E,$3E,$3E
        DC.B    $3F,$3F,$3F,$3F,$3F,$3F,$3F,$3E
        DC.B    $3E,$3E,$3C,$3C,$3B,$3B,$39,$38
        DC.B    $37,$37,$35,$34,$32,$31,$30,$2E
        DC.B    $2E,$2D,$2B,$2A,$29,$27,$26,$24
        DC.B    $23,$22,$20,$1F,$1F,$1D,$1C,$1B
        DC.B    $1B,$19,$19,$18,$18,$16,$16,$15
        DC.B    $15,$15,$15,$15,$15,$15,$15,$15
        DC.B    $15,$15,$15,$15,$16,$16,$16,$18
        DC.B    $18,$18,$19,$19,$1B,$1B,$1C,$1C
        DC.B    $1C,$1D,$1B,$1F,$1F,$1F,$20,$20
        DC.B    $22,$22,$22,$22,$22,$23,$23,$23
        DC.B    $23,$23,$23,$23,$22,$22,$22,$22
        DC.B    $20,$20,$20,$1F,$1F,$1D,$1D,$1C
        DC.B    $1C,$1B,$19,$19,$18,$16,$15,$15
        DC.B    $14,$12,$12,$11,$0F,$0E,$0E,$0D
        DC.B    $0D,$0D,$0A,$09,$09,$07,$07,$06
        DC.B    $04,$04,$04,$03,$03,$03,$01,$01

        DC.B    $01,$01,$01,$01,$00,$00,$00,$00
        DC.B    $00,$01,$01,$01,$01,$01,$01,$01
        DC.B    $03,$03,$03,$03,$03,$04,$04,$04
        DC.B    $04,$04,$06,$06,$06,$06,$06,$06
        DC.B    $06,$06,$07,$07,$07,$07,$07,$07
        DC.B    $07,$06,$06,$06,$06,$06,$06,$06
        DC.B    $06,$06,$04,$04,$04,$04,$04,$04
        DC.B    $04,$04,$04,$04,$03,$03,$03,$04
        DC.B    $04,$04,$04,$04,$04,$06,$06,$06
        DC.B    $07,$07,$07,$0B,$0A,$0A,$0B,$0D
*
*
* NOTE TABLE
*
TABL    DC.W    $0000
        DC.W    $0151,$0166,$017B,$0191
        DC.W    $01A9,$01C2,$01DD,$01FA
        DC.W    $0218,$023B,$0259,$027D
        DC.W    $02A3,$02CD,$02F6,$0323
        DC.W    $0352,$0385,$03BD,$03F3
        DC.W    $042F,$046F,$04B3,$04FA
        DC.W    $0546,$0596,$05EB,$0645
        DC.W    $06A5,$070A,$0775,$07E6
        DC.W    $085F,$08DE,$0965,$09F4
        DC.W    $0A8C,$0B2C,$0BD6,$0C8A
        DC.W    $0D48,$0E13,$0EEA,$0FCD
        DC.W    $10BD,$11BC,$12CA,$13E8
        DC.W    $1517,$165B,$17AC,$1914
        DC.W    $1A92,$1C27,$1DD3,$1F99
        DC.W    $217A,$2378,$2594,$27D0
        DC.W    $2A2E,$2C80,$2F58,$3229
        DC.W    $3524
*
*
* VARIABLE LOCATIONS
*
VOICE1  DS.W    1               LOCATION FOR VOICE 1 POINTER
VOICE2  DS.W    1               LOCATION FOR VOICE 2 POINTER
VOICE3  DS.W    1               LOCATION FOR VOICE 3 POINTER
VOICE4  DS.W    1               LOCATION FOR VOICE 4 POINTER
*
*
* INITIALIZATION PROCEDURE
*
INIT    LEA     WFT(PC),A6      INITIALIZE WAVEFORM POINTER
        LEA     TABL(PC),A5     INITIALIZE NOTE TABLE POINTER
        LEA     MUSTART(PC),A0  GET MUSIC STARTING PLACE
        CLR.B   PORT+$D         TURN OFF HANDSHAKES
        RTS                     END INIT
*
*
* PROCEDURE TO READ NEW DURATION
*
READ.   MOVE.B  (A0)+,D7


        EXT.W   D7
        MULU    $TEMPO,D7  DURATION WILL BE STORED IN D7
        RTS                END READ
*
*
* PROCEDURE TO GET THE NOTES FOR THE FOUR VOICES
*
NOTE    MOVE.B  (A0)+,D1        GET NOTE FOR VOICE 1
        EXT.W   D1              MAKE INTO A WORD
        MOVE.B  (A0)+,D2        GET NOTE FOR VOICE 2
        EXT.W   D2              MAKE INTO A WORD
        MOVE.B  (A0)+,D3        GET NOTE FOR VOICE 3
        EXT.W   D3              MAKE INTO A WORD
        MOVE.B  (A0)+,D4        GET NOTE FOR VOICE 4
        EXT.W   D4              MAKE INTO A WORD
        MOVE.W  0(A5,D1.W),D1   GET INCREMENT FOR VOICE 1
        MOVE.W  0(A5,D2.W),D2   GET INCREMENT FOR VOICE 2
        MOVE.W  0(A5,D3.W),D3   GET INCREMENT FOR VOICE 3
        MOVE.W  0(A5,D4.W),D4   GET INCREMENT FOR VOICE 4
        RTS                     END NOTE
*
*
* PROCEDURE TO PLAY THE MUSIC
*
PLAY    ADD.W   D1,VOICE1       ADD INCREMENT TO WFT POINTER
        ADD.W   D2,VOICE2       ADD INCREMENT TO WFT POINTER
        ADD.W   D3,VOICE3       ADD INCREMENT TO WFT POINTER
        ADD.W   D4,VOICE4       ADD INCREMENT TO WFT POINTER
        MOVE.B  VOICE1,D5       POINT INTO THE WFT
        EXT.W   D5              AND ADD THE BYTES
        MOVE.B  0(A6,D5.W),D0   TOGETHER
        MOVE.B  VOICE2,D5
        EXT.W   D5
        ADD.B   0(A6,D5.W),D0
        MOVE.B  VOICE3,D5
        EXT.W   D5
        ADD.B   0(A6,D5.W),D0
        MOVE.B  VOICE4,D5
        EXT.W   D5
        ADD.B   0(A6,D5.W),D0
        MOVE.B  D0,PORT+$11     OUTPUT THE MUSIC
        SUBQ    D7,PLAY         DECREMENT THE DURATION COUNTER
        RTS                     END PLAT
*
*
* MAIN PROGRAM STARTS HERE
*
MUSIC   BSR     INIT            INITIALIZE
LOOP    BSR     READ            GET DURATION
        BEQ.S   FIN             ALL THROUGH IF ZERO
        BSR     NOTE            GET THE NEXT NOTES
        BSR     PLAY            GO PLAY IT
        BRA     LOOP            CONTINUE LOOPING
FIN     MOVE.B  $60,PORT+$D     TURN HANDSHAKES BACK ON

        MOVE.B  $TUTOR,D7           GO TO TUTOR
        TRAP    #14                 END MUSIC
*
*
* THE MUSIC DATA STARTS HERE
*
```

Update to ELEKTRA!

Package #1	Package #2	Package #3
2 MHz 6809 CPU Board Super Floppy Controller OS-9™ w/Edit. Asm. Debugger	2 MHz 6809 CPU Board Super Floppy Controller 4K Humbug™ and Star-Dos™	2 MHz 6809 CPU Board Super Floppy Controller (No software)
$695.00	**$675.00**	**$550.00**

ELEKTRA OS-9™ with Editor, Assembler, and Debugger **$250.00**
ELEKTRA STAR-DOS™ (Adaptation Guide: $50.00) **$75.00**
OS-9™ Super Modem Program by Epstein Associates **$100.00**

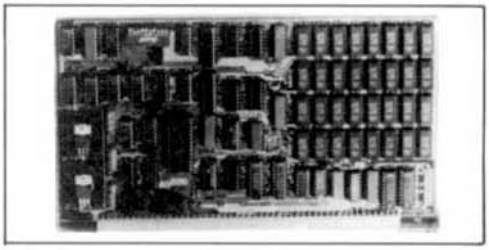
ELEKTRA Super Floppy Controller
(Supports SD, DD, SS, DS, 5", 8", 1MHz, 2MHz, 8 Drives
Emulates the DC-1, DC-2, DC-3, #28, #38, #48, #58 Controllers
Perfect upgrade for the DC-4 **$295.00**

Drivers for TSC's versions of **FLEX™** or **STAR-DOS™** (user installed) **$30.00**
OS-9™ Drivers (Reads and writes CoCo format too, user installed) **$50.00**

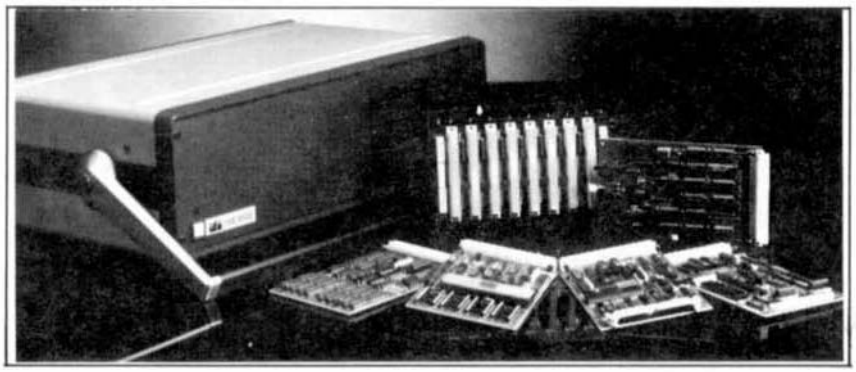
8" Floppy Drive Special w/manual, 90 day warranty
Siemens FDD 100-8 (SSDD) **$135.00**
Siemens FDD 200-8 (DSDD) **$185.00**

Removable Cartridge Winchester Drive (See next page for systems) **$1995.00**

2MHz Memory Boards with on board DAT
by Computer Excellence, Inc.
256K **$749.00** 512K **$1495.00**
1M **$2495.00**



Mizar 68000 — VME Development System with 256K RAM, 360K Floppy,
10 Mbyte Winchester, 4 Serial Ports (synchronous and/or asynchronous),
OS-9, Screen Editor, Assembler, "C" compiler, and SASI interface **$6495.00**



Phone:
AAA Chicago Computer Center
Technical Consultation available most weekdays from 4 p.m. to 6 p.m. CST
(312) 459-0450 120 Chestnut Lane Wheeling, IL 60090
See our catalog and ordering information on the next page.

ELEKTRA COMPUTER SYSTEM Includes chassis, dual port serial interface with two cables, CPU 8/9, 4K Humbug, 56K static RAM, super floppy controller with inboard ribbon cable, Star-Dos, dual 80 track DSDD floppy drives (other combinations available; phone). OS-9 may be substituted for HUMBUG and STAR-DOS. $2795.00

ELEKTRA COMPUTER CABINET THE LARGEST SS-50 COMPUTER CABINET AVAILABLE! Made of heavyweight 0.090" thick aluminum. Interior is 16-1/2" wide by 21-7/8" deep by 6-3/4" high. Heavy duty A.C. line cord. A.C. fuse holder. EMI filter. Fan with filter. Back panel has 10 cutouts for 'D' type data connectors. Front panel has on/off power switch, 2 illuminated push button switches (Reset and NMI/Abort), and two cutouts for 2 full height or 4 half height 5-1/4" disk drives. $250.00

RACKMOUNT ELEKTRACOMPUTER CABINET 17"w x 21.5"d x 6.7"h. Holds one full or two half height 5-1/4" floppy drives. $250.00

Filler Plate for drive opening: $10.00 Fan Filter: $10.00

POWER SUPPLY Highest quality linear Power supply CONSERVATIVELY rated at 15a @ 8v, 3a @ 16v, 3a @ -16v. Multi-tapped primary for fine tuning. $200.00

DISK REGULATOR BOARD WITH CABLES Standard version for 2 floppy drives $50.00. Heavy duty version for 1 Winchester drive and 1 floppy drive or 4 half heights $75.00

AUXILIARY POWER SUPPLY to power second Winchester drive. $125.00

ELEKTRA UNIVERSAL SS-50/SS.50C MOTHERBOARD Heavyweight 0.125" thick, 18" long by 9" wide. 11 memory (50 pin) slots. 8 I/O (30 pin) slots. Complete address decoding and selection, as well as extended address capability, for I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. 1" spacing between all memory and I/O slots. On board baud rate generator with low and high ranges providing jumper selectable rates of 75 through 38,400 for each of the five baud rate lines. Slow device circuitry permitting 1 Mhz 30 pin disk controllers to run with 2MHz 50 pin CPU boards.
Mounting hardware: $5.00 Bareboard w/documentation: $80.00
Assembled w/tin connectors: $380.00 Assembled w/gold connectors: $460.00

ELEKTRA CHASSIS Includes cabinet, 110v power supply, power supply cables, standard disk regulator card with power cables, motherboard with gold square pin connectors, assembled and tested. (Add $25.00 for heavy duty regulator.) $950.00

ELEKTRA 2MHz CPU 8/9 Use either the 6802 or 6808 (to run 6800 software) or 6809. Has provision for up to 3 2716 Eproms, 1K scratchpad, and MC6840 triple timer. Run OS-9™, FLEX™, STAR-DOS™. Bareboard: $50.00 Assembled: $275.00
Optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. $25.00

ELEKTRA OPS DUAL PORT SERIAL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per port. RTS, CTS, DTR, DCD, IRQ, FIRQ/NMI, and baud rate can be appropriately implemented for each port.
Bareboard: $25.00 Assembled: $95.00
Cable with jack socket assemblies (two needed per board) Each: $25.00

ELEKTRA DPP DUAL PORT PARALLEL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request line for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line.
Bareboard: $25.00 Assembled: $80.00
Cable with jack socket assemblies (two needed per board) Each: $25.00

ELEKTRA 64K STATIC RAM/ROM MEMORY BOARDS with gold connectors (tin available) Assembled and tested. With 56K RAM $269.00 With 64K RAM $299.00

ELEKTRA UNIVERSAL SUPER FLOPPY CONTROLLER THE BEST 30 PIN FLOPPY DISK CONTROLLER THAT YOU CAN BUY! Controls up to four 5-1/4" drives and four 8" drives for a total of eight system drives. Single density or double density, 1MHz or 2MHz, 6800 or 6809. (Double density 8" requires 2MHz). Analog phase locked loop data separators with separate adjustments for 5" and 8" drives. Analog write precompensation circuit with separate adjustments for 5" and 8" drives. Designed to meet the data hold requirements of Western Digital floppy controller IC.
Bareboard (experts only): $100.00 Assembled and tested: $295.00
Disk with drivers, setup, and formatting utilities. Specify FLEX 2.0, 6800 Gen. FLEX, FLEX 9.0, FLEX 9.1, 6809 Gen FLEX or STAR-DOS, 5" or 8": $30.00
Disk with drivers for OS-9 (Specify 5" or 8"): $50.00

ELEKTRA WINCHESTER SYSTEMS THE BEST WINCHESTER SYSTEMS THAT YOU CAN BUY! Has automatic error detection and CORRECTION of up to 11 bit burst errors. SS-50 bus, extended addressing capabilities, DMA, on board sector buffer, drivers included for 6809 FLEX, STAR-DOS, or OS-9. Specify whose version of FLEX that you are using. Drivers for 6800 FLEX2 are available for an additional $100.00. Price includes host interface, controller, drive(s), and cables.
7 Megabyte single drive sys. $1995.00 14 Megabyte dual drive sys. $2995.00
12 Megabyte single drive sys. $2295.00 24 Megabyte dual drive sys. $3595.00
19 Megabyte single drive sys. $2995.00 38 Megabyte dual drive sys. $4695.00
(19 Megabyte drives are the largest that can be supported by FLEX)
6 Megabyte removable cartridge single drive sys. $2995.00 Drive only $1995.00
Circuit boards, cables, software (No drives) 995.00
SS-50C DMA Bus Interface board only 695.00

ELEKTRA HD-5 Cabinet for dual 5 1/4" floppy drives with power supply, line cord, fuse, power switch, and power cables to drives. 150.00

ELEKTRA HD-SW As above but with EMI filter, fan, and heavy duty power supply. Powers 1 floppy and 1 Winchester or 4 half height 5" floppies. 199.00

5" ribbon cable for dual outboard 5-1/4" disk drives 40.00
2" ribbon cable for dual inboard 5-1/4" disk drives 35.00
Custom cables available Phone

ELEKTRA HD-8 Dual 8" drive cabinet, EMI filter, fan with filter, power supply and power supply cables. 350.00
Filler plate 10.00
6' ribbon cable for dual 8" disk drives 45.00

ELEKTRA 30 PIN PROTOTYPING BOARD 20.00
ELEKTRA 50 PIN PROTOTYPING BOARD 40.00

GOLD 10 PIN CONNECTORS (Specify male with square pins or female) 1.50
TIN 10 PIN CONNECTORS (Specify male with square pins or female) .50

ELEKTRA is a trademark of AAA Chicago Computer Center.
FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.
HELIX is a trademark of Hazelwood Computer Systems.
HUMBUG, MICROBUG, and STAR-DOS are trademarks of STAR-KITS Software Systems Corp.
OS-9 and BASIC09 are trademarks of Motorola Inc. and Microware Systems Corp.

AAA CHICAGO COMPUTER CENTER (312) 459-0450
120 CHESTNUT LANE • WHEELING, IL 60090
Technical consultation available 4 PM to 6 PM most weekdays. Closed evenings and weekends.

TERMS Minimum order $20.00. Shipping and handling estimates within the Continental U.S., add 3% (MINIMUM $2.50). Illinois residents add 7% sales tax. We will refund your overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINIMUM $10.00). Foreign orders must be prepaid in U.S. dollars. Checks must be drawn on a U.S. bank. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 6 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa, and American Express honored.

Our apology. We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The too frequent changing of our inventory and prices makes it uneconomical to publish a catalog. Our ads are intended to serve that purpose. Prices, specifications, and inventory are subject to change without advance notice.

SUPER MODEM PROGRAM Single character commands. No interrupts required. Transmit manually or transmit disk files (text) of any length to distant computer. Receive and save disk files (text) on local disk system. X-on/X-off supported. Tested for full duplex at speeds up to 9600 baud. Half duplex option. Echo option. Replaces CR with CR/LF (user option). Slow disk file transmit option.
Please specify 6800 or 6809, SSB, STAR-DOS™, or FLEX™, 5" or 8"
Instruction Manual and disk with both source and object code $75.00

OS-9 Super Modem Program by Epstein Associates with auto dial, configuration file, etc. 100.00

ALL IN ONE
Editor — Text Processor — Mailing Labels — Mailing Lists — Multiple Form Letters
Use any CRT terminal and printer — Best Package For The Money Anywhere!
Specify 6800 or 6809, SSB, STAR-DOS™, or FLEX™, 5" or 8" 75.00
Add $35.00 for printed source listing; add $100 for source on disk.
All-In-One, Write'n spell, and Spell'n Fix package 250.00

Software by Technical Systems Consultants, Inc.	FLEX™ Source (List)	FLEX™ Source (Disk)	FLEX™ Man. Only	FLEX™ Object w/Man.	UniFLEX™ Add. Man.	UniFLEX™ Man. Only	UniFLEX™ Object w/Man.
Gen. FLEX w/Edit & ASMB	—	—	—	250	—	—	—
FLEX 9.1 (DC-2) w/Edit & ASMB	—	—	25	150	40	100	550
Advanced Programmers Guide	—	—	25	—	—	—	—
Editor	100	250	25	50	—	—	—
Assembler	150	250	25	50	—	—	—
C bug	175	50	25	75	—	—	—
Extended Basic	—	—	25	100	20	50	200
Basic Precompiler	—	—	25	50	10	25	150
Sort/Merge	—	—	25	75	20	35	150
Utilities	—	Inc.	25	75	10	25	150
Diagnostics	—	—	25	75	—	—	—
Text Processor	150	250	25	75	20	35	150
68000 X-ASMB on 6809	—	—	25	250	20	35	300
Pascal	—	—	50	200	25	50	300
Rel ASMB/Linking Loader	—	—	25	150	20	35	175
6800 X-ASMB on 6809	—	—	—	100	—	—	—
Cobol	—	—	—	—	30	75	750
Fortran 77	—	—	—	—	35	65	450

Software by Microware Systems Corp. (Suggested List Prices, varies/mfg)	Run-Time Package	Source	Manual Only	Object w/Man.
OS-9™ Level I w/Edit, Asm, Debug		400.00	40.00	25000
OS-9™ Level 2 w/Edit, Asm, Debug		400.00	40.00	500.00
OS-9™ Edit, Asm, Debug Pkg			25.00	125.00
Device Driver for Disk Controller (Specify Model)		100.00		
Device Driver for ACIA and PIA		50.00		
Clock Driver for 6840 and 58167 clock chips		35.00		
Entertainment Pack I, or File Handler Toolbox, or NineCom, or Virtual Disk Driver (Level 2 only)			10.00	85.00
Print Spooler (Level 2 only)			15.00	95.00
RMA Relocatable Macro Assembler			20.00	125.00
RMA/68000 Cross Assembler				400.00
BASIC09™ W/Run-Time	50.00	N/A	25.00	200.00
BASIC09 Tour Guide Book			18.95	
"C" Compiler			25.00	250.00
C Programming Language (Kernighan & Ritchie)			19.95	
CIS Cobol Compiler w/Forms 2 Prog. Gen.	50.00	N/A	40.00	400.00
Pascal Compiler	50.00	N/A	25.00	250.00
Sage Application Generator	300.00	N/A	25.00	995.00
Microware yearly support service (All products)				150.00
Edition Update w/o manuals 25.00		Version Update w/manuals		75.00

Special Software

STAR-DOS L1 (Specify ELEKTRA or DC-2, or DC-4)	$75.00	Adaption guide	$50.00
2K MICROBUG 40.00 4K HUMBUG 75.00		Custom versions	$85.00
Spell'n Fix by Peter Stark 178.58		Write'n Spell by Peter Stark	75.11
All-In-One, Spell'n Fix, and Write'n Spell package			250.00
SUPER SLEUTH Disassembler System ($101.00 for OS-9 version)			99.00

SD/DD DISK DRIVES	1 head Tandon	2 heads Tandon	2 heads CDC	1 head MPI	2 heads MPI
30 day guarantee					
5-1/4", 40 tracks	225.00	300.00	300.00	250.00	325.00
5-1/4", 80 tracks	300.00	375.00	375.00	325.00	400.00
MPI or CDC Service Manual (Specify 40 or 80 track)		25.00	Qume DT-8		550.00
Siemens 8" FDD 100-8 (SSDD) $135.00		FDD 200-8 (DSDD)			$185.00

SPECIAL BOARDS
Microtim II Calendar and Clock Board (Assembled) 60.00
Data Mart 16K EPROM bareboard (2708 chips) 30.00

OUTBOARD EPROM PROGRAMMERS BY OPTIMAL TECHNOLOGY
Model EP-2A-79 (Personality modules extra) 169.00
Optimal Technology, Inc. 30 pin parallel I/O board for EP-2A-79 37.00
FLEX™ Software package for EP-2A-79 (Specify 6800 or 6809) 30.00
OS-9 Software package for EP-2A-79 10.00
Model EP-2B-87 (RS-232/20 MA, Motorola Imt, 8K buffer, 1200/9600 baud) 575.00
Model EP-2B-88-4 (Copies 1 to 4 EPROMS) 550.00
Personality/Copy Modules for 2708, 2716, 27C16, 2732, 27C32, 2732A, 2758, MCM68764, MCM68766, 2764, 27C64, 2764A, 27128, 27128A, 27256, 27C256, 2508, 2516, 2532, 2564, 25128, 2816, 2816A, R87C32, 8751, 38E70 $17 to $39

Smoke Signal Broadcasting
DCB-4A Double Density Controller Board for 5" and 8" with DOS 549.00
SCB-69 6809 CPU Board 399.00

GIMIX CLEARANCE SALE	LIST	OUR PRICE		LIST	OUR PRICE
Cable (Par I/O)	24.95	20.00	6800 CPU board	224.03	100.00
80 X 24 Video Boards	398.76	250.00	Single prt ser, 1 cable	113.38	90.00
64 X 16 Video Boards	198.71	100.00	Dual prt ser, 2 cables	138.32	110.00
16K Mem Bds w/cntrl reg		145.00	Systems (trade-ins)		Phone
93L422 DAT chip	17.50	15.00			

HELIX

64K 6809 Computer	$ 395.00	Other computer systems available	
DMA 5" and 8" Floppy Controller	495.00	6809 CPU Board	495.00
68008 board for SS-50	595.00	CP/M-68K	350.00

COMPUTER EXCELLENCE
2MHz Memory board with on board DAT 256K $749.00 512K $1495.00 1M $2495.00

MIZAR
68000, VME development system with 256K RAM, 360K floppy, 10 Mbyte Winchester, 4 serial ports (synchronous and/or asynchronous), OS-9, screen editor, assembler, "C" compiler, and SASI interface $6495.00

SPECIALS

* SSB BFD Floppy Disk Controller (Version 3) Run FLEX or SSB DOS			100.00
* SWTPC DC-4*	230.00	MP-Mb (4K bareboard)	995
* SWTPC DMF-2*	595.00	S-3 * RAM not included	124.50
* SWTPC MP-09 2MHz CPU*	$295.00	*While supplies last	
* High speed tape reader	50.00	300 Baud acoustic modem	129.00
* TI 810 Printer w/lower case and full vertical forms control			1 00.00

WARNING AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled boards. AAA Chicago Computer Center does warranty and maintain service for our assembled boards.

```
MUSTART

DC.W    $181A,$0000,$000C,$1A32
DC.W    $0000,$0C1A,$4000,$0018
DC.W    $1040,$0000,$1810,$323A
DC.W    $000C,$1A00,$0000,$0C1A
DC.W    $0000,$280C,$1A00,$0032
DC.W    $0C1A,$0000,$400C,$1900
DC.W    $0040,$0C10,$0000,$500C
DC.W    $1000,$0052,$0C10,$0000
DC.W    $580C,$1A58,$0000,$0C1A
DC.W    $5800,$400C,$1A58,$004A
DC.W    $0C1A,$5800,$5218,$1052
DC.W    $5800,$1810,$5258,$44A0C
DC.W    $1A00,$0058,$0C1A,$0000
DC.W    $500C,$1A00,$0052,$0C1A
DC.W    $5800,$520C,$1052,$0058
DC.W    $0C10,$5200,$5818,$104A
DC.W    $329C,$0C1A,$0000,$320C
DC.W    $1400,$004A,$181A,$0000
DC.W    $4A18,$1032,$3A40,$0110
DC.W    $0000,$0018,$1032,$3A40
DC.W    $0C1A,$0044,$000C,$1A00
DC.W    $4432,$0C1A,$3A44,$320C
DC.W    $1A3A,$4032,$3010,$323A
DC.W    $400C,$1400,$0000,$0C14
DC.W    $0000,$3218,$143A,$0032
DC.W    $1822,$323A,$440C,$2232
DC.W    $3A40,$0C22,$323A,$440C
DC.W    $1400,$0044,$0C14,$3200
DC.W    $4A18,$1432,$3A4A,$3014
DC.W    $323A,$4A0C,$1000,$0000
DC.W    $0C10,$4000,$0018,$1040
DC.W    $484E,$3014,$4048,$4E0C
DC.W    $1000,$0052,$0C10,$4000
DC.W    $520C,$1040,$0052,$0C10
DC.W    $4048,$4E30,$1E40,$484E
DC.W    $0C0C,$0000,$440C,$0C32
DC.W    $0C44,$300C,$3C00,$4418
DC.W    $0C3A,$0040,$0C1A,$3A44
DC.W    $000C,$1A3A,$4404A,$181A
DC.W    $3A40,$4A30,$103A,$4404A
DC.W    $0C1A,$0000,$4A0C,$1A3A
DC.W    $004A,$181A,$3A40,$4A30
DC.W    $103A,$4404A,$0C1A,$0000
DC.W    $000C,$1A32,$0000,$181A
DC.W    $323A,$0018,$1032,$3A40
DC.W    $0110,$0000,$0018,$1032
DC.W    $1A40,$0C1A,$0044,$000C
DC.W    $1A32,$4400,$0C1A,$3244
DC.W    $3A0C,$1A32,$403A,$3010
DC.W    $323A,$400C,$1400,$0044
DC.W    $0C14,$3A00,$4418,$143A
DC.W    $4A44,$0114,$0000,$0018
DC.W    $223A,$4A44,$0122,$0000
DC.W    $0018,$224A,$4452,$0C14
DC.W    $0000,$520C,$143A,$0052
DC.W    $1814,$444A,$5201,$0044
DC.W    $4A52,$1814,$444A,$5201
DC.W    $1400,$0000,$1814,$444A
DC.W    $520C,$1000,$004E,$0C10
DC.W    $4000,$4E18,$1040,$484E
DC.W    $0100,$0000,$0018,$1E40
DC.W    $484E,$011E,$0000,$0018
DC.W    $1E40,$484E,$0C10,$0000
DC.W    $520C,$1040,$0052,$0C10
DC.W    $4000,$520C,$1040,$484E
DC.W    $301E,$4048,$4E0C,$2400
DC.W    $0044,$0C24,$3C00,$4418
DC.W    $243C,$444A,$0100,$0000
DC.W    $0018,$223C,$444A,$0C1E
DC.W    $3C44,$4E0C,$1E3A,$4404A
DC.W    $0C1A,$3A40,$4A0C,$1A3A
DC.W    $004A,$181A,$0040,$4A18
DC.W    $104A,$3A00,$1810,$4E3C
DC.W    $000C,$1A00,$5200,$0C1A
DC.W    $0052,$3A0C,$1A00,$5240
DC.W    $0C1A,$0052,$4A18,$104A
DC.W    $0052,$0110,$0000,$5218
DC.W    $1040,$4A52,$0C1A,$0052
DC.W    $000C,$1A00,$523A,$0C1A
DC.W    $0052,$400C,$1A00,$524A
DC.W    $0C0A,$4A00,$520C,$0A4A
DC.W    $4E00,$180C,$4A00,$000C
DC.W    $104E,$0000,$0C10,$4E00
DC.W    $3A0C,$104E,$0040,$0C10
DC.W    $4E00,$4818,$1E4E,$4800
DC.W    $181E,$4E48,$400C,$104E
DC.W    $0000,$0C10,$4E00,$360C
DC.W    $104E,$0040,$0C10,$4E00
DC.W    $4818,$1E52,$4800,$181E
DC.W    $5248,$3C0C,$1400,$4A00
DC.W    $0C14,$004A,$320C,$1400
DC.W    $4A3A,$0C14,$004A,$4418
DC.W    $0A44,$004A,$1B0A,$443A
DC.W    $4A0C,$144A,$0000,$0C14
DC.W    $4A00,$3A0C,$144A,$0044
DC.W    $0C14,$4A00,$0018,$4A52
DC.W    $5814,$184A,$5258,$1D0C
DC.W    $0C00,$5C00,$0C0C,$005C
DC.W    $440C,$0C00,$5C4A,$0C0C
DC.W    $005C,$5418,$1A54,$5C00
DC.W    $4181A,$545C,$4A0C,$0C00
DC.W    $5C00,$0C0C,$005C,$440C

DC.W    $0C00,$5C4A,$0C0C,$005C
DC.W    $5418,$1A54,$5C00,$181A
DC.W    $545C,$4A0C,$1A00,$5258
DC.W    $0C1A,$0040,$580C,$1A00
DC.W    $4A52,$0C1A,$0040,$5218
DC.W    $1000,$4052,$1810,$3A40
DC.W    $520C,$1A52,$0000,$0C1A
DC.W    $5200,$3A0C,$1A52,$0040
DC.W    $0C1A,$5200,$4A0C,$0A52
DC.W    $4A00,$0C0A,$4E4A,$0018
DC.W    $0C4A,$0000,$0C10,$484E
DC.W    $000C,$1040,$4E00,$0C10
DC.W    $4800,$520C,$1040,$0052
DC.W    $181E,$4000,$5218,$1E40
DC.W    $4852,$0C10,$0052,$000C
DC.W    $1000,$523C,$0C10,$0052
DC.W    $400C,$1000,$5248,$0C06
DC.W    $4800,$520C,$0648,$004E
DC.W    $180A,$4000,$4A0C,$0C00
DC.W    $4A00,$0C0C,$004A,$3C0C
DC.W    $0C00,$4A40,$0C0C,$004A
DC.W    $4418,$1A44,$4A00,$181A
DC.W    $444A,$3C0C,$2400,$4A00
DC.W    $0C24,$004A,$3C0C,$2400
DC.W    $4A40,$0C24,$004A,$4418
DC.W    $2244,$084A,$181E,$443C
DC.W    $4E0C,$1A4A,$0000,$0C1A
DC.W    $4A00,$3A0C,$1A4A,$003C
DC.W    $0C1A,$4A00,$4018,$104A
DC.W    $4000,$1810,$4A40,$3A0C
DC.W    $1A4A,$0000,$0C1A,$4A00
DC.W    $3A0C,$1A00,$003C,$0C1A
DC.W    $0000,$4018,$1000,$4000
DC.W    $1010,$0040,$3A0C,$1A00
DC.W    $0000,$0C1A,$0000,$320C
DC.W    $1A00,$0036,$0C1A,$0000
DC.W    $3A18,$103A,$4000,$1810
DC.W    $3A40,$320C,$1A00,$4400
DC.W    $0C1A,$0044,$320C,$1A3A
DC.W    $4432,$0C1A,$3A40,$3230
DC.W    $1032,$3A40,$0C14,$0000
DC.W    $000C,$1400,$0032,$1814
DC.W    $3A00,$3218,$2232,$3A44
DC.W    $0C22,$323A,$400C,$2232
DC.W    $3A44,$0C14,$0000,$440C
DC.W    $1432,$004A,$1814,$323A
DC.W    $4A30,$1432,$3A4A,$0C10
DC.W    $0000,$000C,$1040,$0000
DC.W    $1810,$4048,$4E30,$1E40
DC.W    $484E,$0C10,$0000,$520C
DC.W    $1040,$0052,$0C10,$4000
DC.W    $520C,$1040,$484E,$301E
DC.W    $4048,$4E0C,$0C00,$0044
DC.W    $0C0C,$3200,$443D,$0C3C
DC.W    $0044,$180C,$3A00,$400C
DC.W    $1A3A,$4400,$0C1A,$3A40
DC.W    $4A18,$1A3A,$4404A,$3010
DC.W    $3A40,$4A0C,$1A00,$004A
DC.W    $0C1A,$3A00,$4A18,$1A3A
DC.W    $404A,$3010,$3A40,$4A0C
DC.W    $1A00,$0000,$0C1A,$3200
DC.W    $0018,$1A32,$3A00,$1810
DC.W    $323A,$4001,$1000,$0000
DC.W    $1810,$323A,$400C,$1A00
DC.W    $4400,$0C1A,$3244,$000C
DC.W    $1A32,$443A,$0C1A,$3240
DC.W    $3A30,$1032,$3A40,$0C14
DC.W    $0000,$440C,$143A,$0044
DC.W    $1814,$3A4A,$4401,$1400
DC.W    $0000,$1822,$3A4A,$4401
DC.W    $2200,$0000,$1822,$4A44
DC.W    $520C,$1400,$0052,$0C14
DC.W    $3A00,$5218,$1444,$4A52
DC.W    $0100,$444A,$5218,$1444
DC.W    $4A52,$0114,$0000,$0018
DC.W    $1444,$4A52,$0C10,$0000
DC.W    $4E0C,$1040,$004E,$1810
DC.W    $4048,$4E01,$0000,$0000
DC.W    $181E,$4048,$4E01,$1E00
DC.W    $0000,$181E,$4048,$4E0C
DC.W    $1000,$0052,$0C10,$4000
DC.W    $520C,$1040,$0052,$0C10
DC.W    $4048,$4E30,$1E40,$484E
DC.W    $0C24,$0000,$440C,$243C
DC.W    $0044,$1824,$3C44,$4A01
DC.W    $0000,$0000,$1822,$3C44
DC.W    $4A0C,$1E3C,$444E,$0C1E
DC.W    $3A40,$4A0C,$1A3A,$4404A
DC.W    $0C1A,$3A00,$4A18,$1A00
DC.W    $404A,$1810,$4A3A,$0018
DC.W    $104E,$3C00,$0C1A,$0052
DC.W    $000C,$1A00,$523A,$0C1A
DC.W    $0052,$400C,$1A00,$524A
DC.W    $1810,$4A00,$5201,$1000
DC.W    $0052,$1810,$4408A,$520C
DC.W    $1A00,$5200,$0C1A,$0052
DC.W    $3A0C,$1A00,$5240,$0C1A
DC.W    $0052,$4A0C,$0A4A,$0052
DC.W    $0C0A,$4A4E,$0018,$0C4A
DC.W    $0000,$0C10,$4E00,$000C
DC.W    $104E,$0036,$0C10,$4E00
DC.W    $400C,$104E,$0048,$181E
DC.W    $4E48,$0018,$1E4E,$4840
DC.W    $0C10,$4E00,$000C,$104E
DC.W    $0036,$0C10,$4E00,$400C
DC.W    $104E,$0048,$181E,$5248
DC.W    $0018,$1E52,$483C,$0C14
DC.W    $004A,$000C,$1400,$4A32
```

```
DC.W    $0C14,$004A,$3A0C,$1400
DC.W    $4A44,$180A,$4400,$4A18
DC.W    $0A44,$3A4A,$0C14,$4A00
DC.W    $000C,$144A,$003A,$0C14
DC.W    $4A00,$440C,$144A,$0000
DC.W    $184A,$5258,$1418,$4A52
DC.W    $5810,$0C0C,$005C,$000C
DC.W    $0C00,$5C44,$0C0C,$005C
DC.W    $4A0C,$0C00,$5C54,$181A
DC.W    $545C,$0018,$1A54,$5C4A
DC.W    $0C0C,$005C,$000C,$0C00
DC.W    $5C44,$0C0C,$005C,$4A0C
DC.W    $0C00,$5C54,$181A,$545C
DC.W    $0018,$1A54,$5C4A,$0C1A
DC.W    $0052,$580C,$1A00,$4058
DC.W    $0C1A,$004A,$520C,$1A00
DC.W    $4052,$1810,$0040,$5218
DC.W    $103A,$4052,$0C1A,$5200
DC.W    $000C,$1A52,$003A,$0C1A
DC.W    $5200,$400C,$1A52,$004A
DC.W    $0C0A,$524A,$000C,$0A4E
DC.W    $4A00,$180C,$4A00,$000C
DC.W    $1048,$4E00,$0C10,$404E
DC.W    $000C,$1048,$0052,$0C10
DC.W    $4000,$5218,$1E40,$0052
DC.W    $181E,$4048,$520C,$1000
DC.W    $5200,$0C10,$0052,$3C0C
DC.W    $1000,$5240,$0C10,$0052
DC.W    $480C,$064B,$0052,$0C06
DC.W    $4800,$4E18,$0A40,$004A
DC.W    $0C0C,$004A,$000C,$0C00
DC.W    $4A3C,$0C0C,$004A,$400C
DC.W    $0C00,$4A44,$181A,$444A
DC.W    $0018,$1A44,$4A3C,$0C24
DC.W    $004A,$000C,$2400,$4A3C
DC.W    $0C24,$004A,$400C,$2400
DC.W    $4A44,$1822,$4400,$4A18
DC.W    $1E44,$3C4E,$0C1A,$4A00
DC.W    $000C,$1A4A,$003A,$0C1A
DC.W    $4A00,$3C0C,$1A4A,$0040
DC.W    $1810,$4A40,$0018,$104A
DC.W    $403A,$0C1A,$4A00,$000C
DC.W    $1A4A,$003A,$0C1A,$0000
DC.W    $3C0C,$1A00,$0040,$1810
DC.W    $0040,$0018,$1000,$403A
DC.W    $0C14,$0000,$000C,$143A
DC.W    $0000,$1814,$3A44,$4A01
DC.W    $1400,$0000,$1814,$3A44
DC.W    $4401,$1400,$0000,$1814
DC.W    $3A44,$4A0C,$1048,$0000
DC.W    $0C10,$4836,$0018,$1048
DC.W    $3600,$1810,$4A3C,$4401
DC.W    $1000,$0000,$1810,$4E40
DC.W    $4801,$0000,$0000,$0C1A
DC.W    $0000,$520C,$1A40,$0052
DC.W    $011A,$4000,$0018,$1A40
DC.W    $4A52,$011A,$0000,$0018
DC.W    $1040,$4A52,$0110,$0000
DC.W    $0018,$1040,$4A52,$0100
DC.W    $0000,$000C,$1A00,$0052
DC.W    $0C1A,$3A00,$9218,$1A3A
DC.W    $404E,$011A,$3A40,$0018
DC.W    $1E3A,$404A,$0C22,$3A40
DC.W    $4A01,$223A,$4000,$0C22
DC.W    $3A40,$4A01,$0000,$0000
DC.W    $0C24,$0000,$540C,$2444
DC.W    $0054,$0C24,$444A,$5401
DC.W    $2444,$4A00,$0C24,$444A
DC.W    $5401,$0044,$4A00,$1822
DC.W    $444A,$5401,$0044,$4A00
DC.W    $181E,$444A,$5401,$0000
DC.W    $0000,$0C1A,$0000,$540C
DC.W    $1A40,$0054,$011A,$4000
DC.W    $0018,$1A40,$4A52,$0100
DC.W    $0000,$0018,$1840,$003A
DC.W    $0100,$4000,$3A18,$1440
DC.W    $4A3A,$0100,$0000,$000C
DC.W    $1000,$004E,$0C10,$4000
DC.W    $4E18,$1040,$484E,$181E
DC.W    $4048,$5201,$1E00,$0052
DC.W    $0C1E,$4048,$5201,$1E40
DC.W    $4800,$0C1E,$4048,$5201
DC.W    $0000,$0000,$0C10,$0000
DC.W    $4E0C,$1040,$484E,$1810
DC.W    $4048,$4E18,$1E40,$484E
DC.W    $011E,$0000,$0018,$1E40
DC.W    $484E,$0100,$0000,$000C
DC.W    $1400,$0052,$0C14,$4452
DC.W    $0018,$1444,$524A,$0100
DC.W    $0000,$0018,$1444,$524A
DC.W    $0114,$0000,$0018,$1444
DC.W    $524A,$0100,$0000,$000C
DC.W    $1400,$004E,$0C16,$8 C00
DC.W    $4E01,$163C,$0000,$1816
DC.W    $3C46,$4E01,$003C,$4600
DC.W    $0C16,$3C46,$4E01,$163C
DC.W    $4600,$0C16,$3C46,$4E01
DC.W    $1600,$0000,$1816,$3C46
DC.W    $4E01,$0000,$0000,$0C24
DC.W    $0000,$44AC,$243C,$004A
DC.W    $0124,$3C00,$0018,$243C
DC.W    $444A,$0100,$3C44,$000C
DC.W    $223C,$444A,$0122,$3C44
DC.W    $000C,$223C,$444A,$1836
DC.W    $3C44,$4A01,$0000,$0000
DC.W    $0C1A,$3A00,$4A0C,$1A3A
DC.W    $404A,$0C1A,$3A40,$4A01
DC.W    $1A3A,$4000,$0C1A,$3A40

DC.W    $4A01,$003A,$4000,$0C18
DC.W    $3A40,$4A01,$183A,$4000
DC.W    $0C18,$3A40,$4A18,$143A
DC.W    $404A,$0100,$0000,$000C
DC.W    $1000,$004E,$0C10,$4000
DC.W    $4E18,$1040,$4852,$181E
DC.W    $4048,$4E01,$1E00,$004E
DC.W    $181E,$4048,$4E01,$0000
DC.W    $004E,$0C10,$0000,$4E0C
DC.W    $1040,$004E,$1810,$4048
DC.W    $4E01,$0040,$4800,$1810
DC.W    $4048,$4E01,$1E00,$0000
DC.W    $1810,$4048,$4E01,$0000
DC.W    $0000,$0C10,$0000,$540C
DC.W    $1040,$4854,$1810,$4048
DC.W    $5418,$1E40,$4854,$011E
DC.W    $0048,$5418,$1E40,$4854
DC.W    $0100,$0000,$5418,$2840
DC.W    $4654,$1800,$0000,$0018
DC.W    $104A,$3A00,$1810,$4E3C
DC.W    $000C,$1A00,$5200,$0C1A
DC.W    $0052,$3A0C,$1A00,$5240
DC.W    $0C1A,$0052,$4A18,$104A
DC.W    $0052,$0110,$0000,$5218
DC.W    $1040,$4A52,$0C1A,$0052
DC.W    $000C,$1A00,$523A,$0C1A
DC.W    $0052,$400C,$1A00,$524A
DC.W    $0C0A,$4A00,$520C,$0A4A
DC.W    $4E00,$180C,$4A00,$000C
DC.W    $104E,$0000,$0C10,$4E00
DC.W    $360C,$104E,$0040,$0C10
DC.W    $4E00,$4818,$1E4E,$4800
DC.W    $181E,$4E48,$400C,$104E
DC.W    $0000,$0C10,$4E00,$360C
DC.W    $104E,$0040,$0C10,$4E00
DC.W    $4818,$1E52,$4800,$181E
DC.W    $5248,$3C0C,$1400,$4A00
DC.W    $0C14,$004A,$320C,$1400
DC.W    $4A3A,$0C14,$004A,$4418
DC.W    $0A44,$004A,$180A,$443A
DC.W    $4A0C,$1444,$0000,$0C14
DC.W    $4A00,$3A0C,$1444,$0044
DC.W    $0C14,$4A00,$0018,$4A52
DC.W    $5814,$184A,$5258,$100C
DC.W    $0C00,$5C00,$0C0C,$005C
DC.W    $440C,$0C00,$5C4A,$0C0C
DC.W    $005C,$5418,$1A54,$3C00
DC.W    $181A,$545C,$4A0C,$0C00
DC.W    $5C00,$0C0C,$005C,$440C
DC.W    $0C00,$5C4A,$0C0C,$005C
DC.W    $5418,$1A54,$5C00,$181A
DC.W    $545C,$4A0C,$1A00,$5258
DC.W    $0C1A,$0040,$580C,$1A00
DC.W    $4A52,$0C1A,$0040,$5218
DC.W    $1000,$4052,$1810,$3A40
DC.W    $520C,$1A52,$0000,$0C1A
DC.W    $5200,$3A0C,$1A52,$0040
DC.W    $0C1A,$5200,$4A0C,$0A52
DC.W    $4A00,$0C0A,$4E4A,$0018
DC.W    $0C4A,$0000,$0C10,$484E
DC.W    $000C,$1040,$4E00,$0C10
DC.W    $4800,$520C,$1040,$0052
DC.W    $181E,$4000,$5218,$1E40
DC.W    $4852,$0C10,$0052,$000C
DC.W    $1000,$523C,$0C10,$0052
DC.W    $400C,$1000,$5248,$0C06
DC.W    $4800,$520C,$0648,$004E
DC.W    $180A,$4000,$4A0C,$0C00
DC.W    $4A00,$0C0C,$004A,$3C0C
DC.W    $0C00,$4A40,$0C0C,$004A
DC.W    $4418,$1A44,$4A00,$181A
DC.W    $4444,$3C0C,$2400,$4A00
DC.W    $0C24,$004A,$3C0C,$2400
DC.W    $4A40,$0C24,$004A,$4418
DC.W    $2244,$004A,$181E,$443C
DC.W    $4E0C,$1A00,$004A,$0C1A
DC.W    $3200,$4A0C,$1A3A,$004A
DC.W    $0C1A,$4000,$4A18,$1040
DC.W    $004A,$1810,$403A,$4A0C
DC.W    $1A00,$004A,$0C1A,$3A00
DC.W    $4A0C,$1A40,$0000,$0C1A
DC.W    $0000,$4A0C,$1852,$004A
DC.W    $0C18,$4E00,$4A18,$1440
DC.W    $004A,$0C10,$4000,$4E0C
DC.W    $103C,$004E,$0C10,$4000
DC.W    $4E0C,$104B,$004E,$181E
DC.W    $4800,$4E18,$1E48,$404E
DC.W    $0C10,$0000,$4E0C,$103C
DC.W    $004E,$0C10,$4000,$4E0C
DC.W    $1048,$004E,$0C14,$4800
DC.W    $520C,$1448,$004E,$1818
DC.W    $4400,$4A0C,$1A00,$004A
DC.W    $0C1A,$3200,$4A0C,$1A3A
DC.W    $004A,$0C1A,$4000,$4A18
DC.W    $1040,$004A,$1810,$403A
DC.W    $4A0C,$1A00,$004A,$0C1A
DC.W    $3A00,$4A0C,$1A40,$004A
DC.W    $0C1A,$0000,$4A0C,$1852
DC.W    $004A,$0C18,$4E00,$4A18
DC.W    $1440,$004A,$0C10,$0000
DC.W    $4E0C,$103C,$004E,$0C10
DC.W    $4000,$4E0C,$1048,$004E
DC.W    $181E,$4800,$4E18,$1E48
DC.W    $404E,$0C10,$0000,$4E0C
DC.W    $103C,$004E,$0C10,$4000
DC.W    $4E0C,$1048,$004E,$0C10
DC.W    $4800,$520C,$1048,$004E
DC.W    $1810,$4400,$4A0C,$1A00

DC.W    $004A,$0C1A,$3200,$4A0C
DC.W    $1A3A,$004A,$0C1A,$4000
DC.W    $4A18,$1040,$004A,$1810
DC.W    $403A,$4A0C,$1A00,$004A
DC.W    $0C1A,$3200,$4A0C,$1A3A
DC.W    $004A,$0C1A,$4000,$4A18
DC.W    $1A40,$3A4A,$1800,$0000
DC.W    $0000,$0000,$0000,$0000

END     MUSIC
```

```
S11310000D0E0F111214151618191B1C1D1F20226A
S1131010232627292A2B2D2E303132343537383BE0
S1131020393B3B3C3C3E3E3E3E3F3F3F3F3F3F3EE4
S11310303E3E3C3C3B3B393837373534323130 2E39
S11310402E2D2B2A2927262423222201F1F1D1C1B5B
S11310501B191918181616151515151515151515526
S11310601515151516161618181819191B1B1C1CFE
S11310701C1D1D1F1F1F20202222222222232323366
S11310802323232322222222202020 1F1F1D1D1C54
S11310901C1B191918161515141212110F0E0E0D0A
S11310A00B0B0A0808070706060404030303010 1DF
S11310B001010101000000000001010101010101121
S11310C00303030303040404040406060606060695
S11310D006060707070707070706060606060606A5
S11310E006060404040404040404040403030304BB
S11310F004040404040606060707070B0A0A0B0D7D
S1131100000001510166017B019101A901C201DDC9
S113111001FA02180238025902 7D02A302CB02F638
S1131120032303520385 03BB03F3042F046F04B3A7
S113113004FA0546059605EB064506A5070A077554
S11311407E6085F08DE096509F40A8C0B2C0BD648
S1131150 0C8A0D490E130EEA0FCD10BD11BC12CA34
S1131160 13E81517165817AC19141A921C271DD317
S11311701F99217A2378259427D02A2E2CB02F5812
S10711803229352 4B3
S113118C4DFAFE724BFAFF6E41FA00964239000199
S113119C000D4E751E184887CEFC00C04E751218F3
S11311AC488114184882161848831818 4884323514
S11311BC1000343520003635300038354 0004E7578
S11311CCD3781184D5781186D7781188D978118A77
S11311DC1A3811844885103650001A3811864885FF
S11311ECD03650001A3811884885D03650001A3839
S11311FC118A4885D036500013C00001001151CF1C
S113120CFFC04E756100FF7A618A6706619061B018
S113121C60F613FC00600001000B1E3C00E44E4E11
S113122C181A0000000C1A3200000C1A4000 018A6
S113123C1040000018103 23A000C1A0000000C1A6E
S113124C0000280C1A0000320C1A0000400C10008C
S113125C00400C100000500C100000520C10000048
S113126C580C1A5800000C1A5800400C1A58004A12
S113127C0C1A5800521810525800181052584A0C94
S113128C1A0000580C1A0000500C1A0000520C1AC8
S113129C5800520C105200580C1052005818104A96
S11312AC525C0C1A0000520C1A00004A181A000066
S11312BC4A1810323A4001100000001810323A401B
S11312CC0C1A0044000C1A0044320C1A3A44320C26
S11312DC1A3A40323010323A400C140000000C140C
S11312EC00003218143A003218223 23A440C2232DA
S11312FC3A400C22323A440C140000440C143200D0
S113130C4A1814323A4A3014323A4A0C100000008B
S113131C0C1040000018104048 4E301E40484E0C33
S113132C100000520C104000520C104000520C10D3
S113133C40484E301E40484E0C0C0000440C0C32FD
S113134C0044300C3C0044180C3A00400C1A3A444B
S113135C000C1A3A404A181A3A404A30103A404A99
S113136C0C1A00004A0C1A3A004A181A3A404A302D
S113137C103A404A0C1A0000000C1A320000181AD9
S113138C323A001810323A4001100000001810 32A2
S113139C3A400C1A0044000C1A3244000C1A324421
S11313AC3A0C1A32403A3010323A400C14000044D1
S11313BC0C143A00441814 3A4A44011400000185E
S11313DC0000520C143A0052181444 4A52010044AE
S11313EC4A521814444A52011400000018144 44A76
S11313FC520C1000004E0C1040004E181040484E79
S113140C0100000000181E40484E011E0000001888
S113141C1E40484E0C100000520C104000520C1090
S113142C4000520C1040484E301E40484E0C2400D4
S113143C00440C243C00441824 3C444A01000000A1
S113144C0018223C444A0C1E3C444E0C1E3A404AA2
```

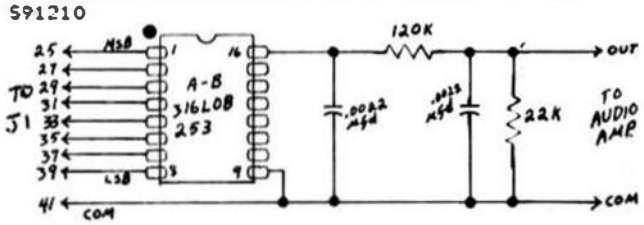
```
S113145C0C1A3A404A0C1A3A004A181A00404A1814
S113146C104A3A0018104E3C000C1A0052000C1A88
S113147C00523A0C1A0052400C1A00524A18104AE4
S113148C005201100000521810404A520C1A00521B
S113149C000C1A00523A0C1A0052400C1A00524A10
S11314AC0C0A4A00520C0A4A4E00180C4A00000C52
S11314BC104E00000C104E00360C104E00400C1058
S11314CC4E0048181E4E4800181E4E48400C104E34
S11314DC00000C104E00360C104E00400C104E0048
S11314EC48181E524800181E5248 3C0C14004A005E
S11314FC0C14004A320C14004A3A0C14004A4418D6
S113150C0A44004A180A443A4A0C144A00000C14BF
S113151C4A003A0C144A00440C144A0000184A526B
S113152C5814184A5258100C0C005C000C0C005C3B
S113153C440C0C005C4A0C0C005C54181A545C00EF
S113154C181A545C4A0C0C005C000C0C005C440C27
S113155C0C005C4A0C0C005C54181A545C00181AED
S113156C545C4A0C1A0052580C1A0040580C1A00BD
S113157C4A520C1A00405218100040521810 3A40AB
S113158C520C1A5200000C1A52003A0C1A52004017
S113159C0C1A52004A0C0A524A000C0A4E4A001801
S11315AC0C4A00000C10484E000C10404E000C105D
S11315BC4800520C10400052181E400052181E4095
S11315CC48520C100052000C1000523C0C100052EB
S11315DC400C100052480C064800520C0648004EB1
S11315EC180A40004A0C0C004A000C0C004A3C0C33
S11315FC0C004A400C0C004A44181A444A00181AAD
S113160C444A3C0C24004A000C24004A3C0C2400A0
S113161C4A400C24004A44182244004A181E443CF4
S113162C4E0C1A4A00000C1A4A003A0C1A4A003C96
S113163C0C1A4A004018104A40018104A403A0C40
S113164C1A4A00000C1A4A003A0C1A00003C0C1AF4
S113165C0000401810004000181000403A0C1A000A
S113166C00000C1A0000320C1A0000360C1A000090
S113167C3A18103A400018103A40320C1A00440040
S113168C0C1A0044320C1A3A44320C1A3A403230D6
S113169C10323A400C140000000C140000321814E0
S11316AC3A00321822323A440C22323A400C22329A
S11316BC3A440C140000440C1432004A1814323A04
S11316CC4A3014323A4A0C100000000C104000004E
S11316DC1810404484E301E40484E0C100000520C5E
S11316EC104000520C104000520C1040484E301E5A
S11316FC40484E0C0C0000440C0C320044300C3CA2
S113170C0044180C3A00400C1A3A44000C1A3A40A3
S113171C4A181A3A404A30103A404A0C1A00004A05
S113172C0C1A3A004A181A3A404A30103A404A0CF9
S113173C1A0000000C1A320000181A323A00181061
S113174C323A400110000000181032 3A400C1A00D2
S113175C44000C1A3244000C1A32443A0C1A32402B
S113176C3A3010323A400C140000440C143A004441
S113177C18143A4A44011400000018223A4A44014D
S113178C2200000018224A44520C140000520C147B
S113179C3A00521814444A520100444A5218144450
S11317AC4A52011400000018144444A520C10000050
S11317BC4E0C1040004E181040484E010000000022
S11317CC181E40484E011E000000181E40484E0CC6
S11317DC100000520C104000520C104000520C101F

S11317FC004418243C444A0100000000 8223C44D4
S113180C4A0C1E3C444E0C1E3A404A0C1A3A404AAE
S113181C0C1A3A004A181A00404A18104A3A00188E
S113182C104E3C000C1A0052000C1A00523A0C1ABE
S113183C0052400C1A00524A18104A00520110006F
S113184C00521810404A520C1A0052000C1A005242
S113185C3A0C1A0052400C1A00524A0C0A4A005212
S113186C0C0A4A4E00180C4A00000C104E00000CD6
S113187C104E00360C104E00400C104E0048181E32
S113188C4E4800181E4E48400C104E00000C104E02
S113189C00360C104E00400C104E0048181E5248D6
S11318AC00181E52483C0C14004A000C14004A3216
S11318BC0C14004A3A0C14004A44180A44004A18FE
S11318CC0A443A4A0C144A00000C144A003A0C1408
S11318DC4A00440C144A0000184A52514184A522C
S11318EC58100C0C005C000C0C005C440C0C005CE0
S11318FC4A0C0C005C54181A545C00181A545C4AB8
S113190C0C0C005C000C0C005C440C0C005C4A0CD1
S113191C0C005C54181A545C00181A545C4A0C1AC7
S113192C0052580C1A0040580C1A004A520C1A0057
S113193C405218100040521810 3A40520C1A5200DF
S113194C000C1A52003A0C1A5200400C1A52004A5B
S113195C0C0A524A000C0A4E4A00180C4A00000C9D

S113196C10484E000C10404E000C104800520C1045
S113197C400052181E400052181E4048520C1000D1
S113198C52000C1000523C0C100052400C1000522F
S113199C480C064800520C0648004E180A40004AEF
S11319AC0C0C004A000C0C004A3C0C0C004A400C79
S11319BC0C004A44181A444A00181A444A3C0C2491
S11319CC004A000C24004A3C0C24004A400C24001D
S11319DC4A44182244004A181E443C4E0C1A4A002D
S11319EC000C1A4A003A0C1A4A003C0C1A4A0040E1
S11319FC18104A400018104A403A0C1A4A00000CBD
S1131A0C1A4A003A0C1A00003C0C1A000040181038
S1131A1C0040001810004 03A0C140000000C143A5A
S1131A2C000018143A444A0114000000 18143A44F3
S1131A3C4A0114000000 18143A444A0C10480000DF
S1131A4C0C1048360018104836001 8104A3C440153
S1131A5C1000000018104E4048010000000000C1A41
S1131A6C0000520C1A400052011A400000181A408F
S1131A7C4A52011A0000001810404A52011000008A
S1131A8C00181040 4A5201000000000C1A000052C9
S1131A9C0C1A3A0052181A3A404E011A3A400018DD
S1131AAC1E3A404A0C223A404A01223A40000C2287
S1131ABC3A404A01000000000C240000540C244459
S1131ACC00540C24444A540124444A000C24444A2F
S1131ADC540100444A00182244 4A540100444A0068
S1131AEC181E444A5401000000000C1A0000540C47
S1131AFC1A400054011A400000181A404A520100BE
S1131B0C000000181840003A010040003A18144034
S1131B1C4A3A01000000000C1000004E0C1040006A
S1131B2C4E181040484E181E404852011E000052D8
S1131B3C0C1E404852011E4048000C1E40485201E5
S1131B4C000000000C1000004E0C1040484E181001
S1131B5C40484E181E40484E011E000000181E40FE
S1131B6C484E01000000000C140000520C144452A6
S1131B7C0018144452 4A010000000018144452 4A3C
S1131B8C011400000018144452 4A01000000000C17
S1131B9C1600004E0C163C004E01163C00001816A4
S1131BAC3C464E01003C46000C163C464E01163C8D
S1131BBC46000C163C464E011600000018163C4616
S1131BCC4E0100000000C2400004A0C243C004A86
S1131BDC01243C000018243C444A01003C44000C01
S1131BEC223C444A01223C44000C223C444A183610
S1131BFC3C444A01000000000C1A3A004A0C1A3A00

S1131C1C4A01003A40000C183A404A01183A400074
S1131C2C0C183A404A18143A404A01000000000CBF
S1131C3C1000004E0C1040004E1810404852181E54
S1131C4C40484E011E00004E181E40484E01000034
S1131C5C004E0C1000004E0C1040004E1810404862
S1131C6C4E0100404800181040484E011E00000070
S1131C7C181040484E01000000000C100000540CD9
S1131C8C1040485418104048541 81E404854011E23
S1131C9C004854181E4048540100000054182840B1
S1131CAC4654180000000018104A3A0018104E3C14
S1131CBC000C1A0052000C1A00523A0C1A00524032
S1131CCC0C1A00524A18104A0052011000005218 03
S1131CDC10404A520C1A0052000C1A00523A0C1AB8
S1131CEC0052400C1A00524A0C0A4A00520C0A4A7E
S1131CFC4E00180C4A00000C104E00000C104E0044
S1131D0C360C104E00400C104E0048181E4E480065
S1131D1C181E4E48400C104E00000C104E00360C91
S1131D2C104E00400C104E0048181E524800181E4D
S1131D3C52483C0C14004A000C14004A320C140097
S1131D4C4A3A0C14004A44180A44004A180A443A01
S1131D5C4A0C144A00000C144A003A0C144A00446D
S1131D6C0C144A000018 4A52581418 4A5258100CB1
S1131D7C0C005C000C0C005C440C0C005C4A0C0C5D
S1131D8C005C54181A545C00181A545C4A0C0C006D
S1131D9C5C000C0C005C440C0C005C4A0C0C005CED
S1131DAC54181A545C00181A545C4A0C1A005258F1
S1131DBC0C1A00405 80C1A004A520C1A00405218C3
S1131DCC100040521810 3A40520C1A5200000C1ACF
S1131DDC52003A0C1A5200400C1A52004A0C0A5285
S1131DEC4A000C0A4E4A00180C4A00000C10484ECB
S1131DFC000C10404E000C104800520C10400052C5
S1131E0C181E400052181E4048520C100052000C70
S1131E1C1000523C0C100052400C100052480C069E
S1131E2C4800520C0648004E180A40004A0C0C009C
S1131E3C4A000C0C004A3C0C0C004A400C0C004AA6
S1131E4C44181A444A00181A444A3C0C24004A0008
S1131E5C0C24004A3C0C24004A400C24004A44182C
```

```
S1131E6C2244004A1B1E443C4E0C1A00004A0C1A18
S1131E7C32004A0C1A3A004A0C1A40004A1B104014
S1131E8C004A1B10403A4A0C1A00004A0C1A3A003C
S1131E9C4A0C1A4000000C1A00004A0C1852004A52
S1131EAC0C184E004A1B1440004A0C1000004E0C3A
S1131EBC103C004E0C1040004E0C1048004E181EE6
S1131ECC48004E1B1E48404E0C1000004E0C103C9E
S1131EDC004E0C1040004E0C1048004E0C144800E0
S1131EEC520C1448004E1B1844004A0C1A00004AAC
S1131EFC0C1A32004A0C1A3A004A0C1A40004A18BE
S1131F0C1040004A1B10403A4A0C1A00004A0C1AA5
S1131F1C3A004A0C1A40004A0C1A00004A0C185297
S1131F2C004A0C184E004A1B1440004A0C100000C9
S1131F3C4E0C103C004E0C1040004E0C1048004E41
S1131F4C181E48004E1B1E48404E0C1000004E0C33
S1131F5C103C004E0C1040004E0C1048004E0C105F
S1131F6C4800520C1048004E1B1044004A0C1A0039
S1131F7C004A0C1A32004A0C1A3A004A0C1A400055
S1131F8C4A1B1040004A1B10403A4A0C1A00004AE9
S1131F9C0C1A32004A0C1A3A004A0C1A40004A1B1D
S1131FAC1A403A4A1B000000000000000000002B
$91210
```



K-BASIC

NEW SOFTWARE

This month I want to tell you about several new and exciting software packages available direct from the manufacturer or S.E. MEDIA.

First, due to several reason, that I will be going into in more depth in months to come, S.E. Media has expanded in the sales and stocking of 68XX(X) software and hardware. Now, S.E. Media is the **LARGEST** 68XX software distributor, IN THE WORLD. We have over 3 0 programs available, for SS50 computers and the CoCo, and more coming each month. **Now that we are into it, WE WILL DO IT RIGHT!** We owe that to you, our readers. You are entitled to a full choice of all the software available for your computers. As time goes by you will see what I mean.

One of the new additions of software suppliers who have asked us to handle their software products is Lloyd I/O (see their ad this issue). So you can buy it direct (which we encourage) or deal with S.E. Media, whichever way you choose. This goes for all the software we catalog. In fact it is our position that they should also advertise and supply their software. After all if it is good enough for us to sell, then they should be sincere enough to advertise and sell it also. Fact is, we give them a good financial reason to advertise and sell their products also (compete with us). **We do not care who else advertises or sells their software,** we just want to insure that it is available to you, and is worthy of your consideration and willingness to buy. For the thousands of you who, for various reasons, prefer to deal with us, well, we will have it!

Most of you are already familiar with the excellent line of macro and cross assemblers from Lloyd I/O. For you who are not, I would heartily recommend that you look into their assemblers, many large users I talk to would use nothing else. You can call them or S.E. Media (use the S.E. Media **FREE WATTS LINE**) for information. If you call S.E. Media ask for Boy Nay - a real expert on most all our (or anybody elses) software products.

K-BASIC

A new Lloyd I/O product is a machine (6809 FLEX™) BASIC binary compiler. K-BASIC takes your BASIC program written in an editor or BASIC and compiles it to a .CMD file. Talk about speed and ease of use! K-BASIC is a BCD math compiler but future versions will allow faster (but less accurate) binary math. Talking about accuracy and math precision - **K-BASIC has 15 digits of precision!** For example a 'real' (floating point) number = +/- 9.9999999999999999+/-99 or 15 printed digits, 9 bytes per real number. Later version which will address various math chips will be even more potent! This is a real plus as most all the compilers that are now available have at the best 6 or 7 digits of precision.

Now - the 'RUMOR' part; If all goes well a special version will be available next spring, it will compile direct TSC Extended BASIC programs to binary .CMD programs. Now that is something we have been needing for quite some time. I will let you know when it is available.

TYPES

Six 'types' of variables are supported: real, no suffix: string, suffix $: byte, suffix ': word, suffix %: long, suffix # and double, suffix (tilde ~). Size of variables is as follows:

real	9 bytes
string	1 byte per char, total 255 per string
byte	1 byte
word	2 byte
long	4 byte
double	8 byte

Note: all are signed notation.

Compiler Fundamentals

Lines are entered with or without line numbers. Line number are required only for GOTO statements, GOTO statements may also reference a label:

```
GOTO 100

GOTO nextsub
```

Either is acceptable.

Comment lines may be either the "'" or the old friend "REM". Blank lines may be used for program line spacing and take up no extra memory assignment space.

LABELS begin in column one and ends with a space. LABELS can be upto 16 characters in length.

Example:

```
100
110
Label
ANOTHERlabel
X
1Xray
```

Upper and/or lower case may be used or mixed. Very long string or expression lines may be concatenated by the use of the backslash. \

Example:

```
IF JUNK$(Bregister%,AREGISTER%)+1=junk0 and JUNK2\
(DREGISTER%,Gregister%)=1 THEN GOSUB 40000 ELSE GOSUB\
LABEL200X
```

NOTE: no space allowed in expressions between the operands and operators. This is not true of most BASIC systems.

In 'expressions' the range of values returned is:

byte	-128 to +127
word	-32768 to +32767
long word	-2,147,483,648 to +2,147,483,647
double long word	-36,028,797,018,963,968 to +36,028,797,018,963,967

CONSTANTS, all constants are converted to their natural form. I.e.: 1.5467, $ef23, $CD56, 4, -8, 2345E65, -$3FED, 2e-06

STRING constants are pretty well as in most all BASIC systems: "PRINT "HELLO WORLD" - "" null string.

Taking the six named types of variable indicated above, and considering three sub-type (non-subscripted, single subscripted and double subscripted) you have effectively 36 types of variable. VARIABLE names can be upto 12 characters in length.

ARRAYS may be established by subscript - DIM JUNKY(3) or by the use of the "AT" statement:

```
AT $E0 4 PORT
```

AT statements fix in memory an absolute memory location. The DIM is assigned a memory location by the assembler, after the program is compiled.

A later and much fuller review of this compiler will be published in a few months. Listed below are some more of the commands, statements, etc. available. Only the less known ones will be touched on in the remained of this column.

START - same as ORG in assembler

STORAGE - a designated place in memory for tables and RMB type storage.

DIGITS - sets the number of digits printed after the dot:

DIGITS=2 ($125.95)

LINE - line length for output - LINE=80

STRING - string lengths, default is 64 - STRING=131 maximum is 255

NAME - program name

TTL - STTL - title and subtitle

PAGE - formfeed

TRON - Trace On - TROFF - Trace mode off

TRACE - turns on trace mode and prints a line similar to:

TRACE MODE AT $XXXX IN LINE n

CODE - ENDCODE - everything between CODE and ENDCODE is for assembly:

```
Example:  CODE
          LEAX 45,U
          LBSR JUNKO
          JSR PCRLF
          ENDCODE
```

LIB <FILE OR PATH NAME> calls other BASIC routines

OOPS!! forgot to tell you, it is also for OS9

ASSEMBLE - command to pass direct to assembler for assembly to a .CMD (FLEX) no extension OS9 program. Provided of course the program compiles without errors.

DIR - tells the compiler which tells the assembler where the output and run time files are located.

Example:

```
FLEX:   DIR 1,0
OS9:    DIR D1/ACC1,/H0/BASIC
```

CHD - changes either work or system drive (FLEX) or directory (OS9)

LET - DATA - DIM - FOR - NEXT - READ - RESTORE - GOTO - GOSUB - RETURN - ON - ON ERROR GOTO - RESUME - IF - PRINT - INPUT - INPUT LINE - END - STOP - POKE - DPOKE - IF THEN ELSE

Disk or channel redirection is by the:

Example:

PRINT #0, CHR$(12); to the system console, by default
PRINT #1, TY$,DISKNAME$,QWERTY(Q$,W$) to a disk file

INPUT #1, STuff$, a$,B,naxto# from a disk file
INPUT LINE #12, MOResTuff$, address$,CSZ$ a line from a disk file

EXEC - calls system utilities and programs as subroutines

```
Example:  EXEC COMMAND$
          EXEC "0.dir"
```

CHAIN - same as EXEC except does not return to calling program

OPEN OLD - opens a file for read, the INPUT - INPUT LINE is used to read from the open channel

Example:

OPEN OLD DATA$ AS 1

OPEN "0.PRINT.SYS" as 0 special case

CLOSE - closes open channels or files

KILL - deletes specified disk file

RENAME - renames specified disk file

SECTOR READ <drive>,<track>,<sector>,<byte array name>

By this statement under FLEX (only) a programmer can easily write programs to read or write ANY disk format

CAR - deallocates memory used by a previously defined DIM statement

Intrinsic Functions

The accuracy of these functions is 15 or so (within 2 or 3 digits) depending on the function, all return 'real' numbers, except where indicated ((WORD)):

EXP - LOG - SQR - COS - SIN - TAN - ATN - ABS - SGN - INT - RND - VAL - PI - STR$ - BSTR$ (BYTE) - WSTR$(WORD) - LSTR$(LONG) - DSTR$(DOUBLE) - LEFT$(string,byte) - RIGHT$(string,byte) - MID$(string,byte1,byte2) - CHR$(byte) - INCH$(BYTE) - ROM$(word) converts to upper or lower case roman numerals - DATE$ system date - BVAL(string) - ASC - LEN - POS - PEEK - INSTR(byte,string1,string2) - CHANNEL 0-12 allowed - WVAL - HEX - DPEEK - WRND - FRE(0) - PTR pointer - ERR - ERL - LVAL -DVAL - wheeee!! (not included)

OSM Compiler

Included is the Lloyd I/O OSM 6809 Macro Assembler. Normal K-BASIC programs call the assembler as:

Example:

ASSEMBLE OSM JUNK.BAS<+ any options>

If the file compiles without error then the assembler is called and the final output is a JUNK.CMD program, ready to call and run from FLEX:

+++ JUNK C/R

A variety of 'ERROR' messages are available from oth the compiler and the assembler.

As there will be a future review into depth concerning this package, I will only say here that OSM is a fine extended macro assembler and is a $99.00 value if purchased alone. It assemble all Motorola type directives as well as many special ones.

THE PRICE

The price for the total package is $199.00

This includes K-BASIC, OSM assembler and a full set of sample programs and LIB type files.

One nice feature is the support of this software. Lloyd I/O not only warranties its software but has a very generous update policy:

No charge the first 60 days. After that a fee of only $15.00 will get you updated to the most recent issue along with a new manual if necessary.

P.S. Our very own Ron Anderson did the math portion, so you know it has to be right!

CRUNCH COBOL

Another new product is CRUNCH COBOL', from England. This product was especially developed for FLEX and supports a large subset of ANSII level 1 COBOL. Many level II features are also included. It supports sequential as well as random and keyed files in FLEX.

Two commands control the system: COBOL compiles a source program - CRUNCH runs the compiled program. CRUNCH COBOL is a one pass compiler with two separate phases. Large programs can be linked or run as a set of overlays.

Regular price $199.00 **SPECIAL $99.00**

Everything mentioned here is available, for immediate delivery from S.E. MEDIA, see their ad this issue.

- - -

BIT BUCKET

GIMIX INC. 1337 WEST 37th PLACE • CHICAGO ILLINOIS 60609 • (312) 927-5510 • TWX 910 221-4055

PRESS RELEASE

GIMIX RECEIVES PRESIDENT'S "E" AWARD FOR EXPORTS

GIMIX Inc. of Chicago received at the National Computer Conference the President of the United States "E" Award for excellence in exporting. Dr. James H. Burrows, Director of the Institute for Computer Sciences & Technology of the National Bureau of Standards made the presentation.

The President's "E" Award was created in 1961 to afford recognition to firms that contribute significantly in the effort to increase U.S. exports. The "E" Awards are made to manufacturers who achieve substantial increases in export sales over a sustained four year period. Export sales must be a significant portion of total sales and be materially in excess of the industry's average percentage in order to qualify for this Award.

GIMIX Inc. started in the Microcomputer industry in 1975 stressing quality and state-of-the-art technology. GIMIX today sells single and multi-user microcomputer systems which are used by software developers, and for test and instrumentation, educational, scientific, and business applications. GIMIX' export sales today account for over 40% of its total revenues. GIMIX successfully competes worldwide and has users on every continent.

For further info, contact:
Richard Ban
GIMIX Inc.
1337 W. 37th place
Chicago, IL 60609



COMPUTER SYSTEMS CENTER
OF ST. LOUIS, INC.
13461 OLIVE BLVD.
CHESTERFIELD, MO. 63017
(314) 576-6020

July 1, 1984

PRESS RELEASE

Computer Systems Center announces the availability of advanced versions of its popular DYNACALC electronic spread-sheet program for FLEX, UniFLEX, OS-9, and CoCo DOS.

DYNACALC is similar in operation to the industry-standard VisiCalc program; the dozens of VisiCalc books available at every neighborhood bookstore are directly usable for learning and applying DYNACALC.

Electronic spread-sheet programs like DYNACALC and VisiCalc bring the computer's power directly to non-programmers. All over the world DYNACALC is being used by thousands of managers, teachers, engineers, accountants, scientists, and other professionals. The versatility and ease of use provided by DYNACALC make it a natural for student and general home applications as well.

DYNACALC retains its highly acclaimed user friendliness while adding such advanced features as logical functions (IF...THEN...ELSE), a random number generator, streamlined printer handling, an improved editor, better screen appearance on terminals that allow cursor suppression, a helpful string locate command, and others.

Users of DYNACALC on standard FLEX, UniFLEX, and OS-9 are invited to upgrade their existing copies. Our update policy is spelled out on the last page of the DYNACALC manual, and provides for free upgrades within 90 days of purchase, $15.00 thereafter.

DYNACALC for TRS-80 Color Computer DOS is a brand new product that brings all DYNACALC's power to Tandy's popular CoCo, but with the major addition of built-in high resolution graphics! CoCo DOS DYNACALC allows you to quicly produce line graphs, vertical bar graphs, even pie charts, all without leaving DYNACALC. You can even copy the graphics screen to your printer, if it has dot-addressable graphics. Best of all, DYNACALC for CoCo DOS is priced at only $99.95. CoCo DOS DYNACALC requires a 64K CoCo and at least one disk drive, but does not need FLEX or OS-9. Dealer inquiries are invited.

DYNACALC Reg. U. S. Pat. Off.
VisiCalc is a trademark of VisiCorp.
FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.
OS-9 is a trademark of Microware and Motorola.
TRS-80 is a trademark of Tandy Corporation

MOTOROLA INC.

Microprocessor Products Division
3501 Ed Bluestein Blvd
Austin, Texas 78721

For further information contact:

Editor Contact: Jim Farrell

(512)928-6572

Reader Contact: Roy Druian

(512)440-2139

MOTOROLA'S NEW MC68020 MICROPROCESSOR
FEATURES FULL 32-BIT DESIGN

NEW YORK/SAN FRANCISCO, JUNE 28, 1984...Motorola, Inc. has formally introduced the MC68020, its most advanced, high-performance, 32-bit microprocessor device.

This new processor is the first true 32-bit design in that it includes non-multiplexed 32-bit internal/external data and address paths--all 32-bit registers, all 32-bit arithmetic and logic units, all 32-bit program counters and all 32-bit stack pointers.

The MC68020 is manufactured using a 2-micron HCMOS process that integrates some 200,000 individual transistors on a 373 x350-mil die that is packaged in a space-efficient, 114-lead, pin grid array package.

The MC68020 operates at a 16.67 MHz clock frequency (60 nanosecond clock period), dissipates less than 1.5 watts, and is completely upward user-object code compatible with earlier members of the M68000 Family. This family of devices has always offered a 32-bit architecture and now provides the full power of 32 bits with the MC68020. In addition, the device includes many new features which fully exploit the complete 32-bit design.

The full 32-bit design structure of the MC68020 provides for direct linear access to 4 gigabytes of logical memory and eliminates instruction timing differences for byte, word and long word operations. The device processes instructions at a sustained rate of 2-3 million instructions per second (MIPS) and at burst rates exceeding 8 MIPS.

The MC68020 fully supports virtual memory and virtual machine concepts. Virtual memory gives each program access to the 4-gygabyte logical address space while allowing the system to contain significantly less physical memory. Virtual machine capabilities refer to the ability to map existing resources and functions of the system onto resources and functions that are required by the user.

The MC68020 provides a means to extend its architecture to off-chip devices through its co-processor interface. This interface allows devices such as the MC68881 floating point co-processor to execute instructions as a part of the main instruction stream.

An important performance feature of the MC68020 is the instruction cache. This 256-byte cache allows simultaneous data and instruction accesses and execution, as well as overall improvement of the performance of the system.

The MC68020 contains a unique bus interface that provides, on a cycle-by-cycle basis, the ability to adjust its data bus width to accommodate 8- or 32-bit devices. The programmer is no longer required to develop programs that are data bus dependent, as the MC68020 will dynamically make the necessary adjustments. Additionally, existing 8- and 16-bit peripheral subsystems can be used with the 32-bit MC68020 by taking advantage of this feature.

The MC68020 also offers extra addressing modes and instructions to further support high-level language development. In addition, 32-bit extensions to existing instructions include an entire family of bit-field operators, double-ended bounds checking, BCD data compression and expansion, module support and enhanced system calling functions.

To efficiently handle task switching, as well as to separate task-related exceptions from system-related exceptions, two system stack pointers are available to the supervisor program. The master stack pointer is associated with the user task so that all task-related exceptions are carried within the user's process control block. When a non-task related exception occurs, the interrupt stack is used for the remainder of system level processing until a new user task is initiated.

To satisfy the requirements of the design-in and development environment, motorola offers cross support under its UNIX™ operating system-derived System V/68™ operating system and its real time VERSAdos TM operating system. These operating systems are available on either the EXORmacs R multi-user development system or the VME/10™ microcomputer system.

A high-performance operating environment is offered in the form of the BENCHMARK 20 TM system, which provides the capability of evaluating the MC68020. This system contains a high-performance CPU board, as well as one megabyte of dynamic RAM. Hardware development is offered in the form of the HDS-400 in-circuit emulator, which may be operated from any number of host systems.

Performance was a major goal of the intense 60-man-year design effort to develop the MC68020, which offers at least a four-times improvement in performance over Motorola's existing 16-bit microprocessors. The MC68020, in following the Motorola technology leadership position, is the true 32-bit performance standard.

UNIX is a trademark of A.T. & T. Bell Laboratories
System V/68, VERSAdos, EXORmacs, BENCHMARK 20 and VME/10 are trademarks of Motorola, Inc.

Mr. Don Williams, Editor
68 Micro Journal
Synoa Cassandra Smith
PO Box 849
Nixon, TN 37343

Donald N. Korte
2714 Thomas
Flint, MI 48504
(313) 238-0718

Don --

Thank you for publishing my PC progress in the June issue of the journal. I should point out some things that have come to my attention since I submitted it.

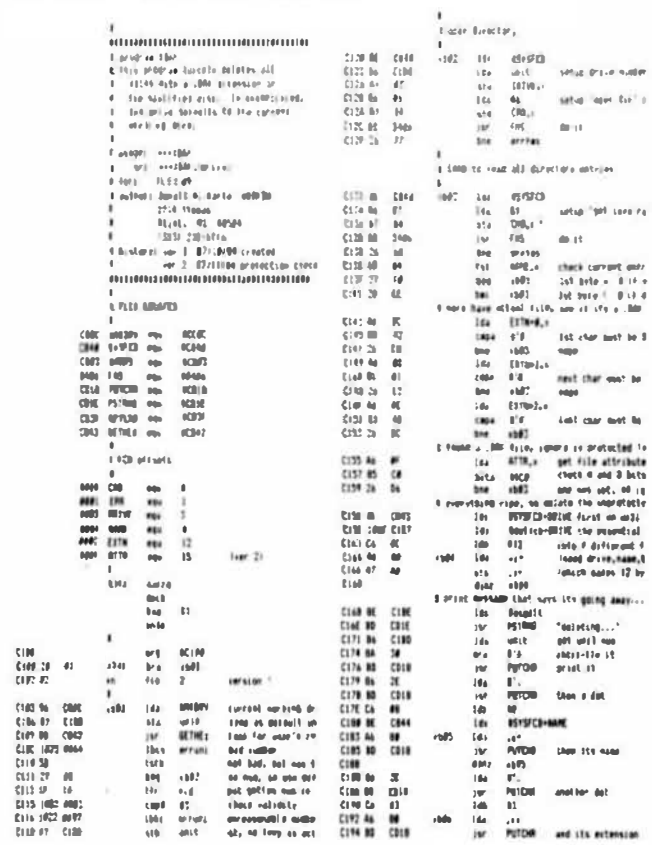
The first thing I should point out is that I typically leave my TTYSET parameters as DP=0 and EJ=0. This makes most programs suppress any sort of paging of their own which would interfere with the paging by PC. I have used some programs which insist on paging (i.e. DMS) by setting the parameters such that the length agrees with what that program likes to work with, and the EJ=0. For example, DMS works ok with a DP=59.

Another thing has to do with the specifics of your printer and printer driver (PRINT.SYS). For my homebrew system, I wrote the printer driver. Since I use the same printer with both my TRS-80 and my 6809 machine, I have setup the driver to ignore LF (0AH) characters, and do a print and linefeed upon receipt of a CR (0DH) character. This makes it work nicely with either machine. In converting PC to run on a more standard machine, I found that I had to change my NULINE equate to "NULINE equ 0AH". In a very kind letter, Bob Roberts observed this problem and suggested adding an explicit linefeed after each requested NULINE. This occurs in the messages, "header" and "head1" and can be accomplished as follows:

```
LNFEED equ $0A
        .
header  fcc NORMAL,/---/,NULINE,LNFEED
        .
headl   fcb NARROW,NULINE,LNFEED,4
```

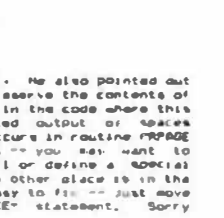
Either of the above changes will work quite well. He also pointed out that the standard TSC printer driver does not preserve the contents of the A register. OOPS! There are two places in the code where this may cause a problem. Both concern the repeated output of spaces without reloading the A register. The first occurs in routine PRPAGE just under the PPOL label. Here I send 3 spaces -- you may want to explicitly load the register between each call or define a special PSPACE subroutine to replace the POUT call. The other place is in the PRDATE routine, in the PRDPL loop. This is easy to fix -- just move the loop re-entry up one line to the "LDA #SPACE" statement. Sorry about that. I should have known better.

On a different subject, please note the enclosed program "XBAK". This is just a short utility I did the other day when I got frustrated when I was cleaning up some disks. While I truly love the automatic backup feature of the TSC EDITOR, I don't really like all those .BAK files cluttering up my disks when I am no longer actively using the associated files. I keep full-disk backups of all my disks also (this is wonderfully easy with Leo Taylor's COPY program), and I especially hate LOADing all those .BAK files to my backup disks. The program works just like IOVI, except that it deletes all files with a .BAK extension. If no drive is specified for XBAK, it will default to the current working drive. The program ignores files that are W or D protected. It logs a message for each file deleted. This may be old hat, but I thought I'd share it anyhow. Hope you like it!



```
PAGE:  2                                FRIDAY     20-JUL-84  06:08:09
program XBAK -- quickly deletes .BAK files
```





Sincerely,

Don Korte

Electronic Specialists, Inc.

171 South Main Street, Natick, Mass. 01760

(617) 655-1532

MODEM PROTECTION

ELECTRONIC SPECIALISTS' expanding product line now includes KLEEN LINE MODEM protection. Models are available for standard 4 pin telephone modular connectors (RJ-11) and wider professional 8 pin connectors (RJ-45).

Intended to suppress software altering telephone and power line spikes caused by lightning, spherics or phone office switch gear, the KLEEN LINE security system uses modern semi-conductor, Metal Oxide Varistor and Gas Discharge Tube suppression techniques.

Model PDS-45P-36A/SUP, modular 8 pin connector, has phone grade suppression on red and green lines (pins 4 & 5) and data grade suppression on yellow and black lines (pins 3 & 6). A 6500 Amp Suppressor protects the AC Power Line. Simple, trouble-free telephone line hook up is provided by the 8 pin modular connector.

FOR MORE INFORMATION: FRANK STIFTER

Measurement Systems Ltd

Press Release

COMMUNICATIONS SOFTWARE FOR OS9

Measurement Systems announces the release of a range of powerful communications software packages for the UNIX type operating system, OS9(c). SyCom(X) provides system to systems communication for both serial and parallel interfaces. Data or files may be transferred in either ASCII or Binary formats. The software operates with any 6809/OS9 computer system equipped with a 6850 serial interface or 6821 parallel interface.

A great deal of attention has been placed on the software transfer protocol to ensure that no data bytes are lost even at transfer rates of 9600 baud on multi-user systems. Software timeouts are incorporated to guarantee that the packages do not "hang" even if the link is disconnected during transfers.

Error codes are reported to the users programme to enable decisions to be made on recovery procedures. The protocol used includes: data transmission checks involving individual byte parity and overall message checksums. Implementation of these packages on OS9 based machines provides a very low cost star network facility for OS9 to OS9 communication. The software may be implemented on Measurement Systems' own OS9 based machine, System96.

OS9 has been designed by Microware for Motorola and is now considered as the industry standard 6809 operating system. It is a UNIX like, multi-user/multi-tasking, real time operating system. It's modular structure makes it adaptable to almost any 6809 computer system. OS9 is widely used for applications for data processing, industrial automation, communications, instrumentation and education.

OS9 is used by a wide range of micro computer manufacturers including, Measurement Systems, Tandy, Dragon, Gimix and SWT.

For more information please contact:

Measurement Systems
78 Faraday Road,
NEWBURY, Berks. RG13 2AD
(0635)42677

RACAL
The Electronics Group

RACAL-DANA
RACAL DANA INSTRUMENTS INC
4 Cambridge Street [illegible] CA [illegible]
[illegible] (714) 859 8999 TWX 910 595 1136 Telex 678 341

RELEASE DATE Immediate CONTACT Bill Mowry

PRESS RELEASE

LOW COST 68000 LOGIC ANALYSIS

IRVINE, CALIFORNIA.....Logic Analyzer Manufacturer Racal-Dana Instruments announced today the introduction of low cost disassembler support for the 68000 16 bit microprocessor. The new option 68K which provides interfacing hardware and software disassembly allows the Racal-Dana Models 202 and 205 48 channel logic analyzers to be easily connected to circuits using the 68000.

The hardware adapter, in addition to providing interconnection between the Racal-Dana logic analyzer and the 68000 automatically establishes the logic analyzer set-up parameters to match the requirements of the 68000 microprocessor. This makes the analyzer/disassembler combination extremely easy to use when debugging 68000 software. Simply connect to the microprocessor and press the "RUN" key.

The software disassembler is contained within EPROM furnished as part of this option. Once installed in the logic analyzer, the EPROM automatically converts the captured data to 68000 mnemonics whenever the hardware adapter is in use. The logic analyzer also provides timing display capabilities in addition to software disassembly.

With the 68000 operating up to its maximum speed of 12.5 MHz, the option 68K can be very selective about the data it collects. There are four different ways data may be qualified:

1) 68000 and DMA or 68000 instructions only
2) Program and/or data
3) Supervisor and/or user
4) Clock qualifier probe (This may be attached to a part of the circuit to qualify recording of the data and may be set to High, low or don't care states.)

There are three external user definable bits for further collection and qualification of data which may be used to measure the response of different parts of the circuit. A cable length of 18 inches from the interface adapter to the chip connection allows access to all microprocessors, even when located in hard to get at places.

The option 68K may also be used with the 68010 microprocessor to allow the reading of the bus cycles.

This new option maintains Racal-Dana's philosophy of designing easy to use, well engineered microprocessor support products. The wide acceptance of the 16 bit 68000 creates the need for economical software disassembly. The Racal-Dana logic analyzer, together with the option 68K meet this growing requirement.

The opt. 68K disassembler sells for $1950.00, and is available 8 weeks ARO. Racal-Dana Instruments is a leading worldwide manufacturer of general purpose test instrumentation for industrial, commercial and government applications.

LPI PROCESSORS INC.
400 TOTTEN POND ROAD
WALTHAM MA 02154
(617) 890-1155
TELEX 95671

LPI FAMILY OF COMPILERS AT A GLANCE

Language Processors, Inc. of Waltham, Mass. today introduced a family of integrated high level language compilers to run on computer systems based on the Motorola MC68000 microprocessor that run the UNIX™ operating system. The compilers are:

- LPI-COBOL 74
- LPI-RPG II
- LPI-Pascal
- LPI-C
- LPI-PL/I
- LPI-BASIC

The compilers have a number of important distinguishing features. They include:

- Polyglot capability, which allows a programmer to mix portions of programs written in different LPI languages into a single program,
- Enhanced reliability, because each compiler is made up of five subsystems, each of which is used in the other compilers,
- High performance, brought about by the fact LPI products are true compilers, and not interpreters.

The compilers use LPI's Component Architecture™, a modular software architecture that combines five standard subsystems: a front end, high level debugger, code generator, global optimizer, and run-time library in each high performance compiler.

The compilers are priced at $50,000 to hardware OEMs, plus royalties based on the size of the system on which they are used. Single copies with run-time license distribution rights are available to qualified software developers.

MICROKEY Ltd
Part of the A.T.S. Group

98a St. James's Street
BRIGHTON
EAST SUSSEX
BN2 1TP

Tel Brighton (0273) 672911

Contact.....Eddie Renouf.

ENGLAND

.....SIGNIFICANT NEW 6809 COMPUTER FROM THE U.K.....

We are currently producing a 6809 single board computer, which we hope will significantly increase the market for FLEX and OS9 software.

The product is a neat, state of the art, 'worst case' design, offering high resolution colour graphics, 80/160 column text and simultaneous yet independent displays on two seperate video monitors (NOT VDU's!) with independent scrolling. This 2MHz computer is plug compatible with Sony 3.5 inch drives and standard 5.25 inch drives. Mixtures of drive types are allowed (eg. two 3.5 inch and one 5.25 inch). On board interfaces for two IBM type keyboards, one UART, and two parallel ports are standard, as is 128K RAM, and 8K of firmware. As an added bonus the machine will also run the 6502 processor with Forth as its primary language and also the 16 bit version of this, which allows 512K of RAM.

By offering hardware with these features at around the same cost as older, more conventional systems, we hope to interest users who would have otherwise purchased CPM ™ other systems. Polyforth with graphic extensions, Pascal and ...siness software will be available from us.

We would be most interested to hear from anyone who wishes to implement high resolution, 16 colour graphics to existing FLEX software or from single board O.E.M. users who require further details on how our system can best benefit them.

A work station using the described single board computer, IBM type keyboard, two 3.5 inch and one 5.25 inch drives will be supplied at the special price of $1899 plus tax for a limited period. Likewise, the single board computer normally costing $549, is now offered at $425.

Finally, we invite enquiries from additional agents or dealers in the U.S.A., Anyone interested should write immediately to us at the above address. Discounts are available.

GIMIX INC 1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609 • (312) 927-5510 • TWX 910-221-4056

GIMIX is proud to announce the availability of

GIMIX RAMdisk for OS-9™

GIMIX RAMdisk is a set of programs that convert a portion of the Random Access Memory (RAM) in an OS-9 level II system into a high-speed, mass-storage device commonly called a RAM or virtual disk.

The "disk" created by GIMIX RAMdisk is used in the same manner as a conventional "physical" disk such as a floppy or hard disk. Files can be created and read or written by including the device name of the RAMdisk in an OS-9 pathlist.

The primary advantage of a "virtual disk" over a conventional disk is its speed. The RAMdisk is considerably faster, because it does not suffer from the mechanical restrictions which limit the access time and data transfer rate of a conventional disk. The access time of the RAM disk is limited only by the speed at which data can be moved by the host processor. In OS-9 GMX III systems, RAMdisk uses the memory-to-memory DMA on the CPU III to achieve the fastest transfer rate possible. The capacity of the RAM disk, and therefore the number and size of the files it can hold, is user-defined; limited only by the amount of free memory in the system.

GIMIX RAMdisk comes on an 8" or 5" (40 or 80 track) diskette and includes: RAMdisk drivers, a utility program to create device descriptors, a utility to remove the RAM disk when it is no longer needed, and a user's manual.

RAMdisk will be included at no charge with all GIMIX OS-9 systems ordered with 256K or more of memory, and is available separately for $125.00

Corrections

Self-modifying Code Unnecessary
For User-entered BASIC Filenames

By Geoffrey A. Gass
5240 S. W. Doach Rd.,
Portland, Oregon 97201

In my article in the May, 1984 issue of '68' Micro Journal I showed an elaborate scheme for using a string variable to provide a user-entered filename for a data file under SWTPC DISK BASIC 3.5.

Bud Hamblen of Nashville points out that the self-modifying code scheme is utterly unnecessary. Bob Uiterwyk's SWTPC DISK BASIC's from 3.0 (MiniFlex) through 3.5 (DMAF-1) or beyond, all allow string variables for filenames, but only if there is no space between the comma after the file number and the variable name. Because of a small bug in the code (omission of a JSR SPSKIP), putting a space after the comma triggers an "illegal filename" error message when GETFIL encounters the "$" of the string variable name. Without the space, the BASIC parser spots the "$" and digs out the actual text of the string variable for GETFIL to chew on.

In the listing in the May issue, then, lines 99-110, 254-295 and 1899-1990 are all unnecessary. Line 1020 should be changed to read:

1020 OPEN #1,N$: REM No space after comma!

The other code pre-processing the N$ variable is still desirable to avoid a program-abort in case N$ does not meet FLEX filename criteria.

Incidentally, BASIC 3.5 supplies the default extension .DAT if no extension is specified. So that portion of the May listing forcing an extension (lines 1740, 1760, 1770 and part of the text in line 1840) may be deleted and line 1780 changed to a simple RETURN.

-o-

For anyone interested in eliminating the BASIC bug, a patch is shown in the listing below. The addresses are valid only for SWTPC DISK BASIC 3.5. The EQU's and ORG will be different for 3.0 and other versions.

There are many "Illegal Character" error routines (ERROR #3) in this section of code; one formerly at $20E7 is deleted and the two branches in GETFN are re-directed to an earlier one. This makes room for the JSR SPSKIP added at $20C4.

Special thanks to Bud Hamblen for disassembling this part of BASIC and discovering the source of the problem.

```
PAGE  001   GETFNPAT

00010                    NAM   GETFNPAT Patch for GETFN Routine
00020                    *     Rev 1.0  July 17, 1984
00030                    OPT   O,NOG

00050                    *Patch allows space between comma and filename
00060                    *in OPEN statement in SWTPC FLEX DISK BASIC 3.5
00080                    *By Bud Hamblen, Nashville TN, and
00090                    *Geoff Gass, Portland, Oregon

00110                    *External References

00130    1F7D    COMCK   EQU   $1F7D    Skip spaces, look for comma
00140    2093    ILLCH1  EQU   $2093    Trigger "ERROR #3" message
00150    0099    SPSKIP  EQU   $0099    Skip spaces; flag EOL
00160    0677    VARFND  EQU   $0677    Look for string variable
00170    0034    POSITN  EQU   $34      Scan pointer(working position
00180    0063    VARPTR  EQU   $63      Points to text of string var.
00190    1F98    SETFN   EQU   $1F98    Call GETFIL, set default ext.
00200    0CE9    ERROR   EQU   $0CE9    Generates ERROR message

00220                    *GETFN routine sets filename for an OPEN statement

00240 20BF               ORG   $20BF

00260 20BF BD 1F7D GETFN JSR   COMCK   Look for comma
00270 20C2 25 CF         BCS   ILLCH1  EOL or no comma?
00280 20C4 BD 0099       JSR   SPSKIP  This is the missing line
00290 20C7 01            NOP           Equalize routine length
00300 20C8 A6 01         LDA A 1,X     Look for 2nd char of argument
00310 20CA 81 24         CMP A #$24    If "$", this is string var.
00320 20CC 26 0E         BNE   FNIN    No "$"? Literal filename.
00330 20CE BD 0677       JSR   VARFND  String variable. Find string.
00340 20D1 29 C8         BVS   ILLCH1  Undefined?
00350 20D3 DF 34         STX   POSITN  Scan pointer
00360 20D5 DE 63         LDX   VARPTR  Set in VARFND
00370 20D7 BD 1F98       JSR   SETFN   Set FCB & call GETFIL
00380 20DA 20 08         BRA   FNERCK  Check for errors
00390 20DC DE 34   FNIN  LDX   POSITN  Probably unnecessary step
00400 20DE BD 1F98       JSR   SETFN   Set FCB, call GETFIL
00410 20E1 09            DEX           SETFN moves XR too far
00420 20E2 DF 34         STX   POSITN
00430 20E4 25 01   FNERCK BCS  ILLFNI  C flags filename error
00440 20E6 39            RTS           All's well
00450 20E7 C6 31   ILLFNI LDA B #$31    Error 31, illegal filename
00460 20E9 7E 0CE9 ERRP39 JMP   ERROR   This pointer called elsewhere

00480                    END

TOTAL ERRORS 00000
```

'68' Micro Journal
P.O. Box 849
Mineola, Tx 75773
U.S.A.

Attn: Don Williams

Dear Don,

Since the publication of MODEM68 in the May 1984 issue of '68' Micro Journal, two bugs have come to light:

```
0182 lda #etaket          should be lda #etaket
```

The second problem is in the close file routine, and causes an incorrect display. The following code will fix the problem, and should be inserted in place of the original * CLOSE FILE #.

```
* CLOSE FILE #
fclo lda a #4 file close
 ldx #fcb
 sta a 0,x
 jsr svfncn get filename into printstring
 lda b #7
 ldx #msg47 'closed'
 stx stloc
 ldx #msg8f
 stx endloc

 jsr svstrg set up rest of string
 lda #msg8a let people know file closed
 jsr putmsg
 lda #fcb
 jsr fas go close file
 rts
```

John Moorfoot
3 Torrence St.
Leopold, VIC 3224
Australia

Editor, 68 Micro:--

In the June issue LaParr Stuart writes: "I dislike the space you give to the 'Color Computer.' It is a useless toy," etc. etc.

Of course the hardware limitations of the Color Computer are both annoying and unnecessary, but it is not really a useless toy for all that. However, the major point deserves an answer, and the answer is this (all figures are taken from the same issue, and the May Color Micro Journal):

"How many SS-50 computers are there out there? They continue to be sold, and I suppose now there might be some 15,000 around." (Ron Anderson's 'Flex User Notes')

"I guess there are over 30,000 OS-9 users now on the CoCo and more coming each day." ('Rumors & Such,' DMW)

"Would you believe that surveys show that less than 15% of the Color Computer Owners OWN a Disk System?" ('This 'n That,' RLN, Color Micro Journal)

I think the figures speak for themselves. The Color Computer is where the old hands are going because even programmers like to eat: and no magazine dedicated to the 68xx community can simply ignore the largest segment of that community.

The people at DataComp, FHL, Spectral, etc., can handle the accusation that CoCo versions of Flex are bastardised. I think you might as well, and as fairly, level the same accusation at any new implementation of CP/M, where the hardware requires a custom BIOS.

Pardon this imprint, the MX-80 went boobs up last week.

Sincerely,

R.W. Odlin

2125 Hoogdal Road
Sedro-Woolley, WA
98284 U.S.A.

GIMIX INC [illegible]

June 28, 1984

Don Williams
'68 Micro Journal
P.O. Box 849
Hixson, TN 37343

Dear Don:

We note that several of your advertisers are using our trademarks without identifying them as trademarks and not properly spelling these trademarks in all capital letters.

GIMIX, GHOST, GMX, and CLASSY CHASSIS are all registered trademarks of Gimix Inc.

We are asking your cooperation in publishing this letter in your next issue.

Very truly yours,
GIMIX, INC.
Richard Don
Richard Don, V.P.

Classified Advertising

TELETYPE Model 43 PRINTER - with serial (RS232) interface, and full ASCII keyboard. LIKE NEW - New cost $1295.00 - ONLY **$759.00** ready to run - Call Tom - Larry - Bob, CPI 615 842-4600

•••

MEX6801 Support (development) system for Exorcisor or Exorterm. Consists of Intercept, Control and Buffer modules, software, documentation. User System Evaluator (USE) capability, real time emulation, EXORbus compatible. List price MEX6801 $2700. For sale at $1200. Also 10 card slot power supply, rack mount chassis (M68MMLC) for $300.
Contact Karl Ritzinger (603)-434-2300 (NH) days.

•••

COMPILER EVALUATION SERVICES
By: Ron Anderson

The S.E. MEDIA Division of Computer Publishing Inc., is offering the following "SUBSCRIBER SERVICE":

COMPILER COMPARISION AND EVALUATION REPORT

Due to the constant and rapid updating and enhanchment of numerous compilers, and the different utility, appeal, speed, level of communication, memory usage, etc., of different compilers, the following services are now being offered with periodic updates.

This service, with updates, will allow you who are wary or confused by the various claims of compiler vendors, an opportunity to review comparisons, comments, benchmarks, etc., concerning the many different compilers on the market, for the 6809 microcomputer. Thus the savings could far offset the small cost of this service.

Many have purchased compilers and then discovered that the particular compiler purchased either is not the most efficient for their purposes or does not contain features necessary for their application. Thus the added expense of purchasing additional compiler(s) or not being able to fully utilize the advantages of high level language compilers becomes too expensive.

The following COMPILERS are reviewed initially, more will be reviewed, compared and benchmarked as they become available to the author:

PASCAL "C" GSPL WHIMISCAL PL/9

Initial Subscription - $39.95
(includes 1 year updates)
Updates for 1 year - $14.50

S.E. MEDIA - CPI
5900 Cassandra Smith, POB 794
Hixson, TN 37343
615 842-4601

Got PC Envy. Selling Gimix 6809 computer with clock, DMA disk controller, 56K RAM. Lots of software including OS-9, FLEX-9. Only $1995. Dual 80tk drives $500.
John Pomeroy (216)372-4457.

•••

Gimix #49 system: 2 40 track dsdd drives
Okidata printer - model U82A
Microware - Pascal, Basic09, Edit, Asm
$2500 or best offer
Call Jim (312)439-6216

•••

SWTPC DMF-2 Dual 8" disk unit + controller, 2.5MB Half list price, $1,495
Mike Dickson (416)292-2850 after 6 p.m. EST

•••

SELL SS50 System 6809 processor, 56K memory, 2x360KBy Disks, PE Terminal, Okidata82A prnter, UDS modem. FLEX9, screen editor, database manager, spreadsheet, communications. C, Pascal, XBasic. $1800/offer.
Doug Beck (415)948-2268

•••

SWTPC 6809 SYSTEM: Dual 8 inch drives, X-12 terminal, 64k RAM, Flex 1.9, several languages. $2700 or best offer.
Jon Kruger, (517)321-4500 or (517)669-9394.

•••

For Sale: SWTPC 6800/2, F&D CPU 6800/09 with both RO moniters, F&D PMB-1 Video, F&D MDI-1 disk contr., 2 SWTPC 8k mem, 2 Gimix 16k mem., 2 Seimens FDD 100 5b drives with p.s. & cabinet. Lots of Flex software. Make an offer on all or part.
414-442-5143, after 7 PM. Jim Bertrand

- - -

68 MICRO JOURNAL PROGRAMS - DISK

Disk-1 Filesort, Minicat, Minicopy, Minifms, **Lifetime, **Poetry, **Foodlist, **Diet.
Disk-2 Diskedit w/ inst.& fixes, Prime, *Prmod, **Snoopy, **Football, **Hexpawn,**Lifetime
Disk-3 Cbug09, Sec1, Sec2, Find, Table2, Intext, Disk-Exp, *Disksave.
Disk-4 Mailing Program, *Finddat, *Change, *Testdisk.
Disk-5 *DISKFIX 1, *DISKFIX 2, **LETTER, **LOVESIGN, **BLACKJAK, **BOWLING.
Disk-6 **Purchase Order, Index (Disk file indx)
Disk-7 Linking Loader, Rload, Narkness
Disk-8 Crtest, Lanpher (May 82)
Disk-9 Datecopy, Diskfix9 (Aug 82)
Disk-10 Home Accounting (July 82)
Disk-11 Dissembler (June 84)
Disk-12 Modem68 (May 84)
DISK-13 *Initmf68, Testmf68, *Cleanup, *Dskalign, *Leobug, Help
Disk-14 *Init, *Test, *Terminal, *Find, *Diskedit, Help

NOTE:
This is a reader service **ONLY!** No Warranty is offered or implied. The Disk Files are as **received by '68' Micro Journal,** and are for reader convenience ONLY (some MAY include fixes or patches). Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other.

PRICE: 8" Disk **$29.95** - 5" Disk **$24.95**

68 MICRO JOURNAL
POB 794
Hixson, TN 37343
615-842-4600

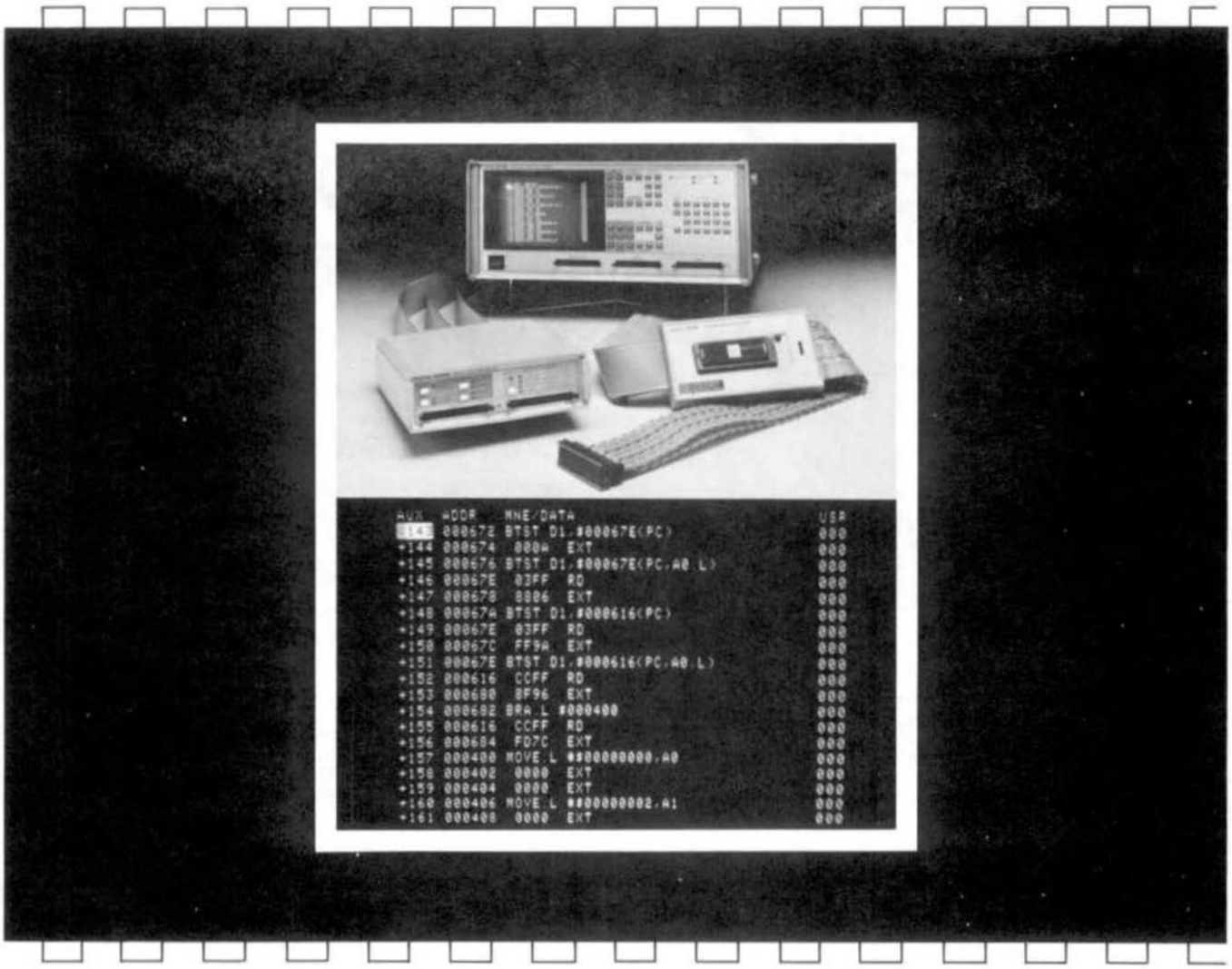
* indicates 6800; ** indicates BASIC SWTPC or TSC
6809 has no indicator.

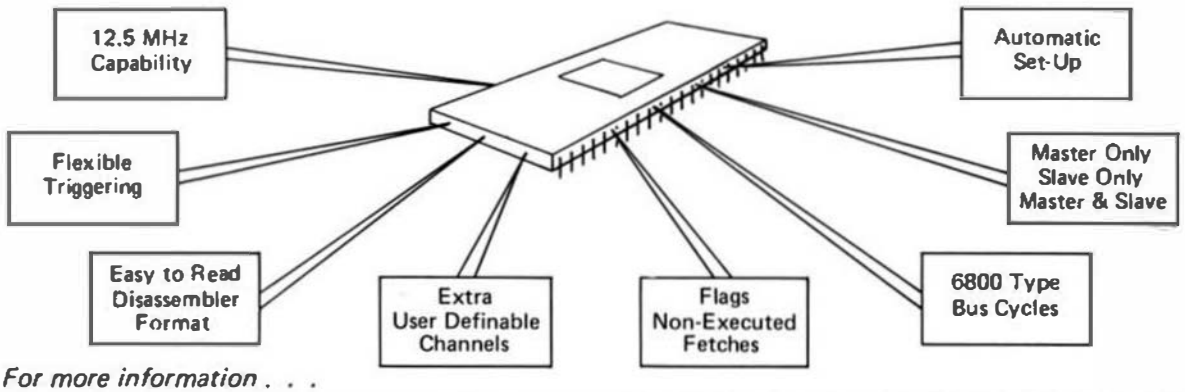
MASTER CARD - VISA accepted
Foreign -- add 10% for surface or 20% for airll

RACAL-DANA

68000 DISASSEMBLER
FOR THE **SERIES 200 LOGIC ANALYZER**

The Window into your Software and Hardware





For more information . . .
Call or write Racal-Dana Instruments, 4 Goodyear St., Box C-19541, Irvine, CA. 92713 (714) 859-8999

GOOD NEWS!



C for the 6809 WAS NEVER BETTER!

INTROL-C/6809, Version 1.5

Introl's highly acclaimed 6809 C compilers and cross-compilers are now more powerful than ever!

We've incorporated a totally new 6809 Relocating Assembler, Linker and Loader. Initializer support has been added, leaving only bitfield-type structure members and doubles lacking from a 100% full K&R implementation. The Runtime Library has been expanded and the Library Manager is even more versatile and convenient to use. Best of all, compiled code is just as compact and fast-executing as ever - and even a bit more so! A compatible macro assembler, as well as source for the full Runtime Library, are available as extra-cost options.

Resident compilers are available under **Uniflex, Flex** and **OS9.**

Cross-compilers are available for **PDP-11/UNIX** and **IBM PC/PC DOS** hosts.

Trademarks:
Introl-C, Introl Corporation
Flex and Uniflex, Technical Systems Consultants
OS9, Microware Systems
PDP-11, Digital Equipment Corp.
UNIX, Bell Laboratories
IBM PC, International Business Machines

For further information, please call or write.

INTROL
CORPORATION

647 W. Virginia St.
Milwaukee, WI 53204
(414) 276-2937

OMEGASOFT INDUSTRIAL STRENGTH PASCAL FOR 6809/68000

OMEGASOFT PASCAL

If you're looking for a language to write real-time process control software, look no further. With the rising cost of labor, it is becoming more critical that a high level language be used. Find out why over 1000 companies have switched to OmegaSoft Pascal for their demanding applications.

WHY PASCAL?

Pascal was designed to teach students how to write structured programs that are easy to read and maintain. In the past decade it has also proved to provide the same advantages in industrial applications.

EXTENSIONS

OmegaSoft has taken the Pascal framework and expanded the basic data types, operators, functions, and memory allocation to fit the needs of real-time systems. These additions fit in the same structure as Pascal and enhance its usefulness without impairing the excellent readability, ease of maintenance, and structured design.

The byte data type allows you to directly address bytes in memory or I/O devices. The common arithmetic operations can be used for bytes along with shift left, shift right, "and", "or", "eor", and complement operators. These operators are also available for integer and hex (2 byte unsigned) numbers.

Longintegers are four byte signed numbers useful for extended range arithmetic commonly needed for machine control. Functions have been added to allow conversion between the various data types. Dynamic length strings allow complex text manipulation and allow effective interactive I/O.

Variables can be placed either on the data stack (default), at an absolute address in memory (for I/O), in base page, relative to the program counter (for constant tables), or defined in another module.

FEATURES

The compiler generates assembly language for assembly and linking to run on the target system. Since a true relocating assembler and linking loader is used, only those runtime modules required are automatically linked in, providing a much smaller object module than other compilers.

Large Pascal programs can be split up into conveniently sized modules to speed the development process. Procedures, functions, and variables can be referenced between Pascal modules and assembly language modules by using Pascal directives.

Full source code is included for the runtime library, the debugger, and other support utilities.

ISO COMPATIBILITY

OmegaSoft Pascal has been tested using the pascal Validation Suite. The Suite is a collection of over 400 Pascal programs designed to test the quality of Pascal Compilers and their runtime systems for compliance with the ISO (International Standards Organization) Pascal standard. OmegaSoft is the only supplier of 6809 native Pascal compilers that publishes this report in its instruction manual.

DEBUGGER

The compiler package includes an interactive, symbolic debugger. The debugger allows setting of breakpoints, displaying and changing variables, and tracing statements. The debugger allows very fast turnaround for programs to be run on the host system.

6809 TARGET SYSTEM

The target system may be any 6809 system. No specific I/O devices are required. The output code is re-entrant and rom-able, perfect for single-board systems up to large development systems. There are no charges for use of the output of the compiler or the object of the runtime library in your products.

6809 HOST SYSTEM

The host system must be 6809 based and have at least 48K of ram (56K recommended) and run one of the following operating systems: MDOS, XDOS, OS—9, or FLEX. Priced from $425.

6809 SUPPORT PRODUCTS

The OmegaSoft Relocatable Assembler and Linking Loader is designed to support the Pascal Compiler Package and can also be used for general assembly language program development. Priced from $125.

OmegaSoft's Screen Editor supports smart terminals and comes complete with the Pascal source. Priced from $90.

For complex real-time applications, the Multi-tasking Kernel provides task scheduling, inter-task communications, and resource interlocking. The Kernel is a runtime library that is accessible as Pascal functions and procedures (with full source included). Priced from $175.

68000 SOFTWARE

A Cross Pascal package is available that runs on a 6809 host system and generates code for a 68000, 68008, or 68010. This package does not include a debugger, but does include a Relocatable Assembler and Linking Loader. Priced from $600.

A Resident 68000 Pascal package is available to run under VERSAdos, with support coming soon for OS-9/68000, CP/M-68K, and UNIX. This package will include the Compiler, Relocatable Assembler, Linking Loader, Debugger, and Screen Editor. Priced from $900.

Dealer and OEM inquiries invited. OmegaSoft products are also available from distributors in Australia and Western Europe, call or write for more information.

OmegaSoft products to run on Motorola development systems are available from Motorola systems distributors in Europe.

OMEGASOFT
CERTIFIED SOFTWARE CORP.
P.O. Box 842
Camarillo, CA 93010
Tel: (805) 987-6426
Telex: 467013

TM: OmegaSoft is a trademark of Certified Software Corporation. MDOS, XDOS, and VERSAdos are trademarks of Motorola. OS-9 and OS-9/68000 are trademarks of Microware. FLEX is a trademark of TSC. CP/M-68K is a trademark of DRI. UNIX is a trademark of A T & T.

FREE DISKETTE WITH EVERY $50 PURCHASE

TOLL FREE 1-800-338-6800 For Ordering

TELEX 558 414 PVT BTH

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343

for information
call (615) 842-4601

CoCo OS-9™ FLEX™

SOFTWARE

ASSEMBLERS

Southeast Media

ASTRUK09

A "Structured Assembler for the 6809" which requires the TSC Macro Assembler. Allows direct use of structured statements such as IF, ELSE, DO, REPEAT, etc., and provides indented level formatting of the listing so that the structure is apparent. Re. '68' Micro Journal, Sept. '83 (program was called "STASM09"; has been renamed due to conflicts).

A User reports

"... I'm very pleased and am now writing almost exclusively in (ASTRUK09). I've selected it over --- for all future s stems development... As (one) of my early evaluations, I rewrote a rather elaborate routine originally done in assembly. Out of the 1000 bytes of code generated, the (ASTRUK09) version used only 20 more bytes than the original. --- could not handle this program since it uses triple-precision fixed point arithmetic... I have a large body of code already written that is incompatible with --- constructs. No problem with (ASTRUK09) and the structure sure helps in understanding the logic!"

F, CCF - $99.95

TSC

Macro Assembler

The FLEX STANDARD Assembler. F,CCF $50.00

Relocating Assembler w/Linking Loader

Use with many of the C and Pascal Compilers. F,CCF $150.00

Great Plains Comp. Co.

RMAC

Relocating, Recursive Macro Assembler and Linking Loader.

F,CCF $120.00; w/Source $240.00

OmegaSoft

[illegible] PRALLI

Relocating Assembler and Linking Loader

F,CCF $125.00; for One Year Maint., add $50.00

Windrush Micro Systems

MACE, by Graham Trott. F,CCF - $98.00

Computer Systems Consultants

SUPER SLEUTH

Computer Systems Consultants Super Sleuth is a "Time Tested", reliable, PROVEN Disassembler that has gained acceptance through out the SS-50 Bus Community as an extremely POWERFUL, INTERACTIVE, Software Tool. The Super Sleuth Software Package consists of 3 Programs: SLEUTH (the Disassembler), CHGLBL (used to globally Change Labels to a meaningful Name), and XREF (a Cross Reference Generator for Source Code Files). SLEUTH will Disassemble Memory Resident 6809 Code and 6800, 6801, 6802, 6803 (the "Baby CoCo"), 6805, 6808, 6809, and 6502 (Apple, Atari, Commodore, etc.) Binary Disk Files. (See Aug. '83 '68' Micro Journal "Color Users Notes" Column for a full Review.)

Color Computer	SS-50 Bus (all w/ Source)
CCD (32K Req'd)	
Obj. Only $49.00	F, $99.00
CCF, Obj. Only $90.00	U, $100.00
CCF, w/Source $99.00	O, $101.00
CCO, Obj. Only $90.00	

ALL Computer Systems Consultants Software runs on the Color FLEX Systems
ALL in stock
call [illegible] for [illegible] DELIVERY

Computer Systems Center

OPTIMIZE +

An "easy to use", powerful Disassembler for Disk Resident 6809 and 6800 Binary Files. Allows the development of a "Control File" of various Program "Boundaries" during successive disassemblies; can use a Label File which automatically replaces a Hex location with a Label Name; includes an XREF Utility; etc. Label Files provided for Mini-FLEX, FLEX2, FLEX9, Color Computer (for use with Color FLEX Systems), etc. OS-9 Version includes special OS-9 options.

CBF, Obj. Only	$100.00
CCO, " "	$59.95
F, " "	$100.00
O, " "	$150.00
U, " "	$300.00

COMPILERS & DECOMPILERS

6809 "Structured" Assembly Lang. Compilers

Windrush Micro Systems

PL/9

By Graham Trott. A combination Editor/Compiler/Debugger, all in ONE PACKAGE; provides a totally INTERACTIVE Program Development Cycle. The Single-Pass Compiler supports large Symbol Names; Variable Types; Pointers; Control Structures (similar to 'C' or 'Pascal'); Stack, A-, B-, and D-Register manipulation; etc. The Source-Oriented Trace/Debugger provides Single Stepping, Break-pointing, etc. An excellent Software Development Tool which provides for the maximum utilization of the power of the 6809.

F, CCF - $198.00

Whimsical Developments

WHIMSICAL

Need the Ease of Design and Maintainability of "Structured Programming" AND the Speed and Control of Assembly Language? Then WHIMSICAL was designed for you! This Single Pass, Recursive Descent Compiler provides the tool for developing simple Utilities to MAJOR Systems in Assembly Language. Supports 3 "Lex" Levels which allow one level of Procedure nesting, or more within "Modules". It is easy to develop programs written for other machines since you are working at the Assembly Language level. Features unified, user-defined I/O; produces ROMable, relocatable, recursive, re-entrant Code; Structured style and statements with Procedures and Modules; supports Byte and Double-Byte primitives with 3 types of Integers (up to 32 bit), Char and Boolean, and unlimited mixed Arrays (vectors only); Interrupt handling; unlimited length Variable Names; Variable Initialization (defaults to $00); Include "Source File" directive; Conditional compiling; direct Code insertion; control of the Stack Pointer; etc. To quote Ron Anderson in his review of WHIMSICAL in the Sept. '83 issue of '68' Micro Journal that, except for the lack of floats, "..., I have to give this one VERY high rating, ...". It is a FAST Compiler which produces FAST Code (his "Prisms" Benchmark ran at 9 secs. on a 2 MHz System).

F and CCF - $195.00

'C' Compilers

Windrush Micro Systems

C Compiler

By James McCosh. Full featured C Compiler for the FLEX Operating System (lacking ONLY "bit-fields"), including an Assembler. Requires the TSC Relocating Assembler IF the user wishes to implement his own Libraries.

F and CCF - $295.00

Introl

C Compiler

A full-featured C, streamlined for the 6809. Generates very efficient object code. Output "benchmarks" close to 10MHz 68000 in 8 Bit Operations; 1.5 times faster than a 4 MHz Z80 when using a 2MHz 6809 System (Re. p 43, "68" Micro Journal, May '83). Floats, etc.

F, CCF, and O - $375.00
U - $425.00
One Year Maint. - $100.00

VISA MasterCard

TOLL FREE 1-800-338-6800 For Ordering

SOUTH EAST MEDIA

5900 Cassandra Smith Rd. Hixson, TN 37343

info (615) 842-4601

CoCo OS-9™ FLEX™ SOFTWARE

Availability Legends —
F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

"FLEX is a trademark of Technical Systems Consultants
"OS9 is a trademark of Microware

PASCAL Compilers

TSC

PASCAL Compiler

Native Code Compiler (UCSD Oriented).

F and CCF - $288.88

Lucidata

PASCAL Compiler

P-Code Compiler (ISO Standard). Designed especially for Microcomputer Systems; Run-time System checks available resources for each task, allowing operation on even minimal computer systems. Allows linkage to Assembler Code for maximum flexibility.

F and CCF 5" - $198.00
F 8" - $285.00

OmegaSoft

PASCAL Compiler

For the PROFESSIONAL; ISO Based, Native Code Compiler. Primarily for Real-Time and Process Control applications. Use custom I/O devices in place of the Pascal INPUT and OUTPUT; Long Int. (32 Bit); Dynamic length strings; Interrupt processing, ROM-able, PIC, Re-Entrant Code, etc. POWERFUL Includes Source for the Symbolic Debugger, Runtime, and several Utilities. Requires a "Motorola Compatible" Relocating Assembler and Linking Loader.

F and CCF - $425.00
One Year Maint. - $100.00

DECOMPILERS

Southeast Media

DUB (A UniFLEX "basic" De-Compiler)

Re-Create a Source Listing from UniFLEX Compiled basic Programs. EASY to Use; works w/ ALL Versions of UniFLEX basic; Output to Disk or Terminal. Time TESTED and PROVEN: SOLID!

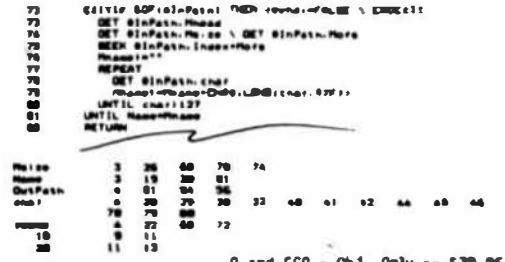
U - $219.95

UTILITIES

Southeast Media

BasicO9 XRef

This BasicO9 Cross Reference Utility is a BasicO9 Program which will produce a "pretty printed" listing with each line numbered, followed by a complete cross referenced listing of all variables, external procedures, and line numbers called. Also included is a Program List Utility which outputs the listing without the overhead of building the cross reference table, which allows it to run considerably faster when only a "pretty printed" listing with line numbers is desired. Requires BasicO9 or RunB for operation.



O and CCO - Obj. Only -- $39.95
O and CCO - w/ Source - $79.95

Southeast Media

OS-9 VDisk

Give your OS-9 Level I System the speed of memory access that can be several orders of magnitude over your present floppy disk drive. Use that Extended Memory capability of your SWTPC or Gimix CPU card (or any other that has the same format DAT). The size of the Virtual Disk is completely variable in whole increments of 4K up to 960K, which is all that these systems can address beyond the base page that OS-9 Level I uses. By putting all of your CMDS Directory on your Virtual Disk, you can have the fastest execution speed possible (next to eating up System Memory with all of them). You can also set up high speed inter-process communications via random virtual disk files and not eat up valuable system memory with pipe buffers. Some Assembly Required - Level I ONLY.

O, obj. only - $79.95
w/ Source - $149.95

FREE DISKETTE WITH EVERY $50 PURCHASE

TOLL FREE 1-800-338-6800
For Ordering
TELEX 558 414 PVT OTH

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343
for information
call (615) 842-4601

CoCo OS-9™ FLEX™
SOFTWARE

Southeast Media

O-F

---- OS/9 to FLEX - FLEX to OS/9 ----

Finally; the barrier has been removed between OS/9 and FLEX formatted disks! Now you can READ from, and WRITE to, a Single Sided 5" or 8" FLEX diskette from OS-9 with O-F. O-F is a new and unique program, written in BASIC09 (with Source), that performs the following functions;

REFORMAT: A BASIC09 Program that reformats a chosen amount of an OS-9 disk to FLEX Format so it can be used normally by FLEX.

FLEX: A BASIC09 Program that does the actual read or write function to the special O-F Transfer Disk, all selected from a user-friendly menu. Functions provided include reading the FLEX Directory, Deleting FLEX Files, Copying both directions, etc. All selections are interactive and complete, including all necessary prompts to the operator.

FLEX users can read, write and use the special disk as any other FLEX disk, provided the FLEX directory is not allowed to continue beyond track zero (too many files).

F and CCF - $79.95

Southeast Media

COPYVAULT

--- Copy LARGE Disks to several smaller disks ---

The following FLEX utilities allow the backup of ANY size disk to any SMALLER size diskettes (Winchester to 8's or 5's, 8" to 5's, etc.). By simply inserting diskettes as requested by COPYVAULT, a large disk system may be downloaded to your present floppy disk system, any size. No need to fiddle with directory deletions or any of the other tedious operations that must be done using the normal copy routines.

COPYVAULT.CMD understands normal "copy" syntax and always keeps up with files already copied by maintaining directories for both host and receiving disk system, eliminating hours of tedious keyboard entries and other time consuming cleanup chores.

BACKUP.CMD is a special program that downloads "random" type files, any size.

RESTORE.CMD a special program to restructure copied "random" files for copying, or recopying back to the host system.

FREELINK.CMD a "bonus" utility that "relinks" the free chain of floppy or hard disk thereby eliminating fragmentation.

Completely documented source files included.

ALL 4 Programs (8" or 5") $99.50

Southeast Media

CHESS 6809

Requires FLEX and DISPLAYS On Any Type Terminal

Features:

*Four levels of play.
*Swap side. *Point scoring system.
*Two display boards. *Change skill level.
*Solve Checkmate problems in 1-2-3-4 moves.
*Make move and swap sides. *Play white or black.

This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1600+ (better than most 'club' players at higher levels).

F and CCF - $79.95



"FLEX is a trademark of Technical Systems Consultants
"OS9 is a trademark of Microware



Availability Legends —
F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

FREE DISKETTE WITH EVERY $50 PURCHASE

☎ TOLL FREE 1-800-338-6800 For Ordering

TELEX 558 414 PVT BTH

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343

for information
call (615) 842-4601

CoCo OS-9™ FLEX™

SOFTWARE

Southeast Media

DIET-TRAC Forecaster

DIET-TRAC Forecaster is an XBASIC program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C P GT) or grams of carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individual are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's weight for a 30-day period is printed.

F - $59.95
U - $89.95

Southeast Media

XDATA
A COMMUNICATION Package
for the UniFLEX Operating System

Allows UniFLEX Based Systems to Transmit and Receive files to and from other Computer Systems via Modem. Use with CP/M, Main Frames, other UniFLEX Systems, etc.

-- Verifies Transmission integrity using checksum or CRC
-- Automatically Re-Transmits bad blocks
-- Transmits data in 128 byte blocks

U - $299.99

Southeast Media

JUST
Text Formatter

JUST, a Text Formatter developed by Ron Anderson, provides numerous features which make it a valuable addition to any FLEX Users Software Library. JUST is designed for formatting Text Output for Dot Matrix Printers and provides many unique features:

- Output the "Formatted" Text to the Display for format analysis and change.
- Output the "Formatted" Text to a Text File for use with the supplied FPRINT.CMD for producing multiple copies of the Text on the Printer INCLUDING IMBEDDED PRINTER COMMANDS (this Utility is very useful at other times also, and worth the price of the program by itself).
- "User Configurable" for adapting to other Printers (comes set up for Epson MX-80 with Graftrax); provides for up to ten (10) imbedded "Printer Control Commands", such as Italics on and off, boldface on and off, etc.
- Automatic compensation for a "Double Width" printed line.
- Includes the normal line width, margin, indent, paragraph, space, vertical skip lines, page length, page numbering, centering, fill, justification, etc.
- Use with ANY Editor.
- Supplied with "Structured Source" (Windrush PL/9); easy to see the flow of the program.

F and CCF - $49.95

Lucidata

PASCAL UTILITIES
Requires LUCIDATA Pascal ver 3.

XREF -- produce a Cross Reference Listing of any text; oriented to Pascal Source.

F and CCF - $25.00

INCLUDE -- allows the inclusion of other files in a Source Text; has unlimited nesting capabilities. Also allows Binary File inclusions.

F and CCF - $25.00

PROFILER -- produces an Indented, Numbered, "Structogram" of a Pascal Source Text File. Allows viewing the overall structure of large programs, and provides clues as to the integrity of the program. Supplied as Source Code; requires compilation.

F and CCF - $25.00

Lucidata

COPYCAT
Pascal NOT required

Allows reading TSC Mini-FLEX, SSB DOS68, and Digital Research CP/M Disks while operating under FLEX 1.0, FLEX 2.0, or FLEX 9.0 with 6800 or 6809 Systems. COPYCAT will not perform Miracles, but, between the program and the manual, you stand a good chance of accomplishing a transfer. Includes Utilities to List Directories, Copy Files, and convert Text Files when required. Also includes a Utility for investigating Physical Compatibility problems. Programs supplied in Modular Source Code (Assembly Language) to make it easier to solve unusual problems.

F and CCF 5" - $50.00
F 8" - $65.00

Computer Systems Consultants

FLEX DISK UTILITIES

Eight (8) different FLEX Utilities that should be a part of every FLEX Users Toolbox; Assembly Language (Source Code):

Copy a File with CRC Errors, so it can possibly be salvaged; Test Disk for errors; Compare two Disks; a fast Disk Backup Program; Edit Disk Sectors; Linearize Free-Chain on the Disk; print Disk Identification; and Sort and Replace the Disk Directory (in sorted order).

F and CCF - $50.00

WORD PROCESSORS

Alford and Associates

SCREDITOR III

EXTREMELY Powerful Screen-Oriented Editor/Word Processor. Almost 50 different commands; EXCELLENT Documentation (over 300 pages), including a full Tutorial Section to help you learn how to use the system. Features Cursor-based editing, dynamic Screen Formatting (what you see is what you get), Multi-Column display and editing, "decimal align" columns (AND add them up automatically, if wanted), define multiple keystroke macros, even and odd page number headers and footers, imbed printer control codes in text, full justification series of commands, full "help" support, store common command series on disk for future use, etc. Easy "Set-Up" (for example, you just hit the key you want to use for a specific function, such as "cursor up", and the System reads an stores that key - no digging into tech manuals for codes, etc.); use supplied "set-ups", or remap the keyboard to what you are used too. Except for proportional printing, this package will DO IT ALL!

6800 or 6809 FLEX or SSB DOS, OS-9 - $175.00

Great Plains Computer Co.

STYLOGRAPH

A full-screen oriented WORD PROCESSOR -- (now runs on the Data-Comp and FHL Color FLEX Systems; uses the 51 x 24 Display Screens). Full screen display and editing (i.e., what you see is what you get); supports the Daisy Wheel proportional printers.

SPECIAL CCF - $195.00

F and D - $295.00 U - $395.00

SPELL

Fast Computer Dictionary.

F, CCF, OS/9 - $125.00 U - $175.00

MAIL MERGE

Greatly extends the power and flexibility of STYLOGRAPH.

F, CCF, D - $145.00 U - $195.00

VISA MasterCard

*FLEX is a trademark of Technical Systems Consultants
™OS9 is a trademark of Microware

☎ TOLL FREE 1-800-338-6800 For Ordering

SOUTH EAST MEDIA

5900 Cassandra Smith Rd. Hixson, TN 37343

CoCo OS-9™ FLEX™

Info (615) 842-4601

SOFTWARE

Availability Legends —
F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

Great Plains Computer Co.

MAIL MERGE

Greatly extends the power and flexibility of STYLOGRAPH. Allows Multiple Text files to be printed out as one large document. Provides for merging information into the Text File during printing (such as different names and addresses), etc.

F, CCF, O - $145.00
U - $195.00

SouthEast Media

SPELLB "Computer Dictionary"

OVER 120,000 words!

No more "Let your fingers do the walking through the Dictionary" while you are entering Text with your favorite Editor or Word Processor. SPELLB is more than just "another Spelling Checker"; it allows you to look up a word from within your Editor or Word Processor so that you KNOW it is right WHEN YOU TYPE IT IN with the SPH.CMD Utility (which operates in the FLEX Utility Space). Yes, it ALSO allows you to check and update the Text after you are finished; along with allowing you to ADD WORDS to the Dictionary, "Flag" questionable words in the Text for evaluation later, "View a word in context" before changing or ignoring, etc. SPELLB first checks a "Common Word Dictionary", then the normal Dictionary, then a "Personal Word List", and finally, any "Special Word List" you may have specified. SPELLB also allows the use of Small Disk Storage systems.

F and CCF - $129.95

Great Plains Computer Co.

SPELL

Fast Computer Dictionary -- allows directly changing the Text File, adding words to the dictionary, etc. 75,000 words in less than 400 sectors.

F, CCF, OS/9 - $125.00
U - $175.00

DATA BASE MANAGEMENT SYSTEMS

Southwestern Applied Business Systems

XDMS

Possibly one of the most powerful Database Management Systems available, this machine language program is small enough to operate on a **single sided 5" disk**, yet provides the **speed** of M.L. and **power** limited only by the user's imagination. This DMS supports Relational, Sequential, Hierarchical, and Random Access File Structures, and has Virtual Memory capabilities for those Giant Data Bases. XDMS Level I provides a functional "entry level" System which provides for defining a Data Base, entering and changing the Data, and producing Reports. **XDMS Level II** adds the POWERFUL "**GENERATE**" facility which uses an English Language Command Structure in manipulating the Data to create new File Structures, Sort, Select, Calculate, etc. **XDMS Level III** adds several special "Utilities" which provide additional ease of working with the various structures, changing System Parameters, etc.

XDMS Lvl I - F & CCF - $129.95
XDMS Lvl II - F & CCF - $199.95
XDMS Lvl III - F & CCF - $269.95
XDMS System Manual only - $24.95

Great Plains Computer Co.

INFOMAG DBMS

An XBASIC, Menu Driven, DBMS with "Built-In" Audit Tracking, Extremely Powerful Report & Format Capabilities, etc. This **Time Proven** DBMS will become the "Work Horse" of your Software Stable.

F and CCF $295.00
U $395.00

ACCOUNTING PACKAGES

Great Plains Computer Co. and **Universal Data Research, Inc.** both have Business Packages written in TSC XBASIC for FLEX, **CoCo FLEX**, and **UniFLEX** ----

* * * * * * * * * * * *

FREE DISKETTE WITH EVERY $50 PURCHASE

TOLL FREE
1-800-338-6800
For Ordering

TELEX 558 414 PVT BTH

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343

for information
call (615) 842-4601

CoCo OS-9™ FLEX™

SOFTWARE

Computer Systems Consultants

BASIC UTILITY PROGRAMS

Ten BASIC Programs to:

A BASIC Resequencer with EXTRAs over "RENUM"; works with ALL Versions of FLEX BASIC **AND** the Precompiler, checks for missing label definitions, processes Disk to Disk instead of in Memory.

Compare, Merge, or Generate Updates between two BASIC Programs, check BASIC Sequence Numbers, compare two unsequenced files, and 5 Programs for establishing a **Master** Directory of several Disks, and sorting, selecting, updating, and printing paginated listings of these files.

A BASIC Cross-Reference Program, written in Assembly Language, which provides an X-Ref Listing of the Variables and Reserved Words in TSC BASIC, XBASIC, and **PRECOMPILED** BASIC Programs.

ALL Utilities include Source (either BASIC or Source Code). An EXCELLENT Value!

F and CCF - $25.00
UniFLEX - $50.00

Computer Systems Consultants

FULL SCREEN INVENTORY/MRP

The Full Screen Inventory System provides a means of maintaining small inventories. Using a linked, keyed random file structure based upon the item field, it keeps the file in alphabetical order for easier inquiry. With the FIND command, the user may locate and/or print all records matching on partial or complete item, description, vendor, or attributes. Items in backorder or below minimum stock levels may be located and/or printed thru the same process. Printed output may be produced in item or vendor order. A materials requirement planning (MRP) capability for manufacturing environments is included to allow the maintenance and analysis of Hierarchical assemblies of items in the inventory file. It requires TSC's **Extended BASIC.**

F and CCF - $100.00, U - $150.00

The Virginia Company ***Bizpack***

BIZPACK is used for storing accounting, numeric, and financial data which can then be used for planning, budgeting, forecasting, analyzing, etc. While "Electronic Spreadsheets" are extremely useful in many situations, BIZPACK excels in businesses where there are numerous expense columns, revenue sources, significant business indicators, large numbers, erratic week-to-week and month-to-month fluctuations, etc. BIZPACK helps determine statistical relationships, establish trend lines, "smooths" data via moving averages, analyze seasonal data, adjusts for inflation, lags data in Statistics or Column functions, plots data, etc. BIZPACK is oriented toward time series analysis of businesses. The Program displays information on the screen in Columns of Information with each Row conforming to a defined Period of Time (weeks, months, years, etc.), and is very easy to use (data is easy to enter, change, and modify; commands can be renamed to suit the users requirements; unlimited ability to create specialized commands using common BASIC Statements; etc.). Requires TSC's **Extended BASIC.**

F and CCF - $135.00
with Source - $250.00

***** SPECIAL *****

Purchase XBASIC and BIZPACK together for $223.50
-- a Savings of $13.50 --

VISA MasterCard

*FLEX is a trademark of Technical Systems Consultants
*OS9 is a trademark of Microware

TOLL FREE 1-800-338-6800 For Ordering
SOUTH EAST MEDIA
5900 Cassandra Smith Rd. Hixson, TN 37343
Info (615) 842-4601
CoCo OS-9™ FLEX™ SOFTWARE

Availability Legends —
F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

FREE DISKETTE WITH EVERY $50 PURCHASE

TOLL FREE 1-800-338-6800 For Ordering

TELEX 558 414 PVT BTH

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343

for information
call (615) 842-4601

CoCo OS-9™ FLEX™

SOFTWARE

*** SPECIAL ***

Purchase XBASIC and XDISPACK together for $221.50
— a Savings of $13.50 —

Computer Systems Consultants

TABULA RASA SPREADSHEET

TABULA RASA is similar to DESKTOP/PLAN and provides for the generation and maintenance of tabular computation schemes often used for analysis of business, sales, and economic scenarios. Its menu-driven user interface provides these capabilities even to those users with no programming experience. Its extensive report-generation capabilities allow the user to generate professional results with minimum effort. It requires TSC's Extended BASIC.

F and CCF - $100.00. U - $125.00

Computer Systems Center

DYNACALC

THE Electronic Spread Sheet for 6809 Computer Systems. An extremely POWERFUL Business Tool, this Program will find an unlimited number of "non-business" applications, also (for example, a Full Junior College Electronics Curriculum was set up using DYNACALC). Advanced features like "Table Lookup" make Income Tax work easy; Column or Row Sorting for numerous applications; etc. Completely "Memory Resident", Machine Language, this Program is FAST. Provides STANDARD FLEX Text File output for use with BASIC, Word Processors, Pascal, "C", etc. Also available for Data-Comp and FHL FLEX systems using the 80 x 24 Displays.

F and SPECIAL CCF - $200.00
CoCo DOS - $99.95
O - $250.00
U - $237.00

ODDS & ENDS

Computer Systems Consultants

FULL SCREEN FORMS DISPLAY

This package supports any Serial Terminal with cursor control of Memory-Mapped Video Displays. The package substantially extends the screen Input/Output capabilities of TSC's Extended BASIC programs by providing a simple, table-driven method of describing and using full screen displays. These table entries are easy to set up and maintain, and are normally stored on disk and read as required. A simple, interactive means of generating the forms and the data field definitions is provided.

F and CCF - $50.00. U - $75.00

Computer Systems Consultants

FULL SCREEN MAILING LIST

The Full Screen Mailing List System provides a means of maintaining simple mailing lists. Using a random fill structure based on the first character of the name field, it maintains the file in alphabetical order for easier inquiry. With the FIND command, the user may locate all records matching on partial or complete name, city, state, zip, or attributes. Printed listings and output to labels may also be produced on the same selective basis. It requires TSC's Extended BASIC.

F and CCF - $100.00. U - $110.00

COLOR COMPUTER SOFTWARE

Stearns Electronics

FORTH

Intrigued by FORTH??? Here is a FORTH package tailored to the Color Computer! This package is supplied on Tape, with instructions for transferring it to disk if you wish. Written primarily in machine language, it's speed is unparalleled. A full Semigraphic-8 Editor is provided, along with "goodies" like Graphics and Sound Commands, Printer Commands, Auto-Repeat and Control Keys, etc. If you are interested in Learning FORTH, a Trace Feature is provided which is invaluable. If you are a FORTH Pro, this package provides CPU carry Flag accessibility, Fast Task Multiplexing, Clean Interrupt Handling, etc. (Or: you won't "out grow" the Basic capabilities of this implementation). Combine this package with Leo Brodie's EXCELLENT Book "Starting FORTH", and you will be a FORTH Expert before you know it (and have a lot of fun doing it!).

Color Computer TAPE - $58.95

Custom Software Engineering, Inc.

Color Computer **GRAPHIC SCREEN PRINT** Programs

Dumps any "PMODE" Screen to the Printer with the BASIC USR Function. Shift the Printout Left or Right or Reverse Print (Dark for Light Screen and Vice Versa). All Programs on Tape.

GSPR	for R.S. LP-VII/VIII & CMP 100/200/400	$7.95
GSPR2	for Epson w/ Graftrax and Graftrax +	$9.95
GSPG	for Gemini 10 and 15	$9.95
GSPRP	for the Prowriter Printers	$9.95

Custom Software Engineering, Inc.

DATE-O-BASE CALENDAR Program

A Menu Driven EXTENDED BASIC Program which allows the entry of up to 12 Memos per Day, each of which may contain up to 28 Characters, for any day of the Month between the years 1700 and 2099. A Graphic Calendar shows which days contain Memos, and a "Key Word" Search is provided which can be output to the Screen or Printer.

TAPE DATE-O-BASE CALENDAR
(Each Tape File will hold up to 400 Memos) $16.95

DISK DATE-O-BASE CALENDAR
(4,000 Memos at 300/Month per Disk) $19.95

Custom Software Engineering, Inc.

That's INTEREST-ing

Interested in INTEREST (the Money Kind)? An EXTENDED BASIC Program that will help you deal with numerous problems requiring interest calculations. Present Value, Rate of Return, Current Bond Yield and Rate of Return to maturity, Loan Repayment Amortization Schedules, etc.

TAPE - $29.95

Custom Software Engineering, Inc.

DISK DATA MANAGER 64K

An EXTENDED BASIC Data Management System w/ Mach. Lang. Routines. Allows a max of 246 Chars. and 14 Fields per Record, and another Record can be linked to the first; 8 Char. Field Names, up to 99 Chars. per Field. Powerful On-Screen editor for input and update, flexible Output capabilities including output to Disk Files for use by other Programs. Change File Definition without re-entering the Data, Split Files, etc. Allows Multiple Field Sorts, Select on any combination of Fields, etc. An extremely POWERFUL TOOL. Instructions provide examples of Mailing Lists and a Financial Stock Profit and Loss Tracking System.

DISK - $54.95

Custom Software Engineering, Inc.

DISK DOUBLE ENTRY

DISK EXTENDED BASIC Accounting Program w/ Mach. Lang. Routines. A "Traditional" Accounting Package for Small Business, Clubs, Churches, Personal Use, etc. Up to four levels of subtotals with Trial Balance, Income Statement, and Balance Sheet Reports. DDE allows up to 300 accounts and a Trial Balance of $9,999,999.99. Transactions may be up to 14 lines long, and comments and explanations may be freely used. Accounts are traceable to the journal transaction, which may include comments. Screen reports allow review of past transactions and current balances.

DISK - $44.95

VISA MasterCard

*FLEX is a trademark of Technical Systems Consultants
*OS9 is a trademark of Microware

TOLL FREE 1-800-338-6800 For Ordering

SOUTH EAST MEDIA

5900 Cassandra Smith Rd. Hixson, TN 37343 Info (615) 842-4601

CoCo OS-9™ FLEX™ SOFTWARE

Availability Legends —
F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

Computer Systems Center

DYNASHARE

— Multi-User, Multi-Tasking with FLEX —

Southeast Media is now shipping DYNASHARE **FROM STOCK** - the multi-user, multi-tasking capability of **DYNASHARE** allows FLEX users the advantages of more sophisticated and time saving computer usage without having to buy or learn a new language or Operating System syntax. **DYNASHARE**, as its name implies, allows true "time-sharing" operation under the popular FLEX operating system, and also allows each user to run two simultaneous jobs (multi-tasking); even on single-user systems. For example, while in EDIT, you can list another file or examine a directory. Or, you might look up an item in a Data Base while a Sort is in progress! **DYNASHARE** also provides some fringe benefits that will be greatly appreciated by FLEX users, including type-ahead, command line editing, and instant response to "escape".

DYNASHARE is the painless method! Use your existing Flex computer by simply adding 64K of RAM for each user and/or task. Fact is, you still use FLEX just like you always have! **DYNASHARE** is not intended as competition to UniFLEX. It does not improve on the speed of FLEX, and does not offer password protection or other niceties of a full-blown multi-user system. What **DYNASHARE** does do is give FLEX users a **low-cost way to use existing software in a multi-user, multi-tasking environment**, so your existing FLEX versions of BASIC, XBASIC, editors, assemblers, disassemblers, sort/merge packages, word processors, compilers, DYNACALC spread-sheet package, and so on are still good.

NOTE -- The **initial release** of DYNASHARE is for **SWTPC S/09** Computers, but versions will also be available for other popular extended-memory (up to **1024K**) systems, such as HELIX and GIMIX. A minimum of **128K** of **RAM** will be required with **ALL** versions. **DYNASHARE** requires **64K of RAM for each active task**; thus a 256K system could allow foreground-background operation on two terminals, or foreground-only operation on four terminals.

AVAILABLE NOW from **Southeast Media** - **$200.00**

AUTHORS - PROGRAMMERS
QUALITY SOFTWARE NEEDED
FLEX - UniFLEX - OS/9 - Color Computer

For the past several months, we at the **Southeast Media Division** of **Computer Publishing, Inc.** (CPI), the parent company of '68' **MICRO JOURNAL** and **COLOR MICRO JOURNAL**, have debated expanding our software distribution business. Many other magazines have been doing so for years (in fact, MOST were in the Software Distribution Business BEFORE they began to publish a Magazine). Presently there are many fine examples of software that has been developed by YOU, our readers, that will never see the "light of day" due to the **Cost of Advertising** and **TIME and Cost** involved in the **production, distribution,** and **Customer SUPPORT** of that software unless SOMEONE, with enough exposure and the willingness to continually advertise, runs with the ball.

Software is the "backbone" for the REAL utilization of any Computer System, and ours are no exception! This has been no simple decision. While we realize that there could be some conflict with some of our advertisers, we ALSO hear a LOUD and CONTINOUS cry for **HELP** from our Readers. From day one, the **foremost concern** of '68' MICRO JOURNAL has been it's **READERS**! Therefore, our **Southeast Media Division** will accept, for appraisal for possible Distribution, 6809 software; Games, Utilities, Software Development, Business Application Programs, etc.

FREE DISKETTE WITH EVERY $50 PURCHASE

☎ **TOLL FREE** TELEX 558 414 PVT BTH

1-800-338-6800

For Ordering

SOUTH EAST MEDIA

5900 Cassandra Smith Rd.
Hixson, TN 37343

for information
call (615) 842-4601

CoCo OS-9™ FLEX™

SOFTWARE

In the past there has been **too much** software offered that was not quite ready. We will strive to eliminate that element. But, right up front, we tell you only that we will do our very best; nothing more. Also, we will strive to keep cost to a bare minimum, while securing for the author a fair return in royalty payments, promptly paid, and in customer support for his product.

Of course, we will expect, no -- **DEMAND**, that the author keep the product free of errors (bugs), and maintain it in a prompt and business like manner. Also we shall require that authors be willing to furnish 'source' for those programs that justify, by price and utility, inclusion of same. The lack of source code, properly commented, is a continual complaint we hear. Not all programs will be sold with source, but where necessary, we will insist that it be included.

In some instances the program may be small or short and not justify itself as a "single" sale product. In this event it will be combined with other like programs, and offered as a package. In that event, the royalties will be split between the various authors.

If you have software that you feel will qualify under this program, please contact one of the people below. Remember, if your software has any problems or "funnies" — **GET IT STRAIGHT BEFORE YOU CONTACT US!!** Also get your source code in proper shape and well commented; there is too much 99% code already drifting around.

If your software is **READY** contact:
Bob May, Don Williams, or Tom Williams

Southeast Media is a division
of **Computer Publishing, Inc.** (CPI),
a family of 100% 68XX support facilities.

VISA

MasterCard

*FLEX is a trademark of Technical Systems Consultants
**OS9 is a trademark of Microware

☎ TOLL FREE
1-800-338-6800
For Ordering
SOUTH EAST MEDIA
5900 Cassandra Smith Rd.
Hixson, TN 37343
info (615) 842-4601
CoCo OS-9™ FLEX™
SOFTWARE

Availability Legends —
F = FLEX, CCF = Color Computer FLEX
O = OS-9, CCO = Color Computer OS-9
U = UniFLEX
CCD = Color Computer Disk
CCT = Color Computer Tape

COMPARE

our EPROM PROGRAMMER with the field.

All data taken directly from manufacturer's current advertising. Software, interfaces, or personality modules may also be required at additional cost.

- Triple voltage power
- Supplied in kit form

	A	B	C	D	E	F
INTERFACE	S30	PAR	PAR	SER	S30	SER	SER
INTELLIGENT	NO	NO	NO	YES	NO	YES	YES
PROGRAMS							
2704*			●				●
2508	●		●	●	●	●	
2708*			●	●	●		●
2758	●	●	●	●		●	
2516	●	●	●	●	●	●	
2716	●	●	●	●	●	●	●
2716*			●		●		●
2532	●		●	●	●	●	●
2732	●		●	●	●	●	●
2732A	●		●	●	●	●	●
2564	●		●	●	●	●	●
2764	●		●	●	●	●	●
2528	●		●	●			
27128	●		●	●			
2616							●
68764			●				
8748					●		
8749					●		
TOTAL	11	3	12	6	11	11	11
PRICE	$125	$45*	$169	$289	$375	$489	$575

EPRAM EPROM Programmer, $125. Personality module for 2508, 2758, 2516, and 2716 included. Specify CPU, disk size, and operating system (TSC's FLEX or SSB's DOS) when ordering. Manual only, $10 refundable with EPRAM purchase.

UNITEK • P.O. Box 671 • Emporia, VA 23847



NEW!

Compact Flexible 6809 Computer

The new ST-2900 system — a complete 64K small business or hobbyist computer is only one of its many possible configurations. Among its features are:

- Small enough to hold in your hand! (Eurocard size: 3.9" x 6.3")
- Two board "system" for greater versatility than single board computers.
- CPU Board — powerful 6809E processor, 16K or 64K RAM, 2K-8K EPROM, 2 RS232 serial ports with software programmable baud rates, 16 bit counter/timer. Run the CPU board all by itself, or plug your own custom board or our FDC board into the expansion connector.
- FDC Board — double-sided/double-density floppy disk controller with adjustment free digital data separator and write precompensation, 2-8 bit parallel ports, 2-16 bit counter/timers, prototyping area.
- Runs FLEX/9 OS-9 under development (trademarks of Technical Systems Consultants, and Microware, respectively).
- Available as bare PC boards or partially assembled boards. All have solder mask both sides plus silkscreened component overlay.
- FLEX Conversion Package consists of disk/console/printer drivers, disk formatting utility, and related utilities.
- **This month's special** includes the ST-2900 CPU and FDC boards (bare boards with documentation), an EPROM (with ST-MON monitor program), and the FLEX Conversion Package, all for only **$95** (US funds) plus $5 shipping/handling ($10 overseas). Canadian residents $129 in Canadian funds plus $4 s/h — B.C. residents add 7% tax. Price valid until Oct. 31/84.

Send money order to:

Write for free brochure and complete price list.

(604) 255-4485
(4 - 5 pm Pacific Time)

SARDIS TECHNOLOGIES

2261 E. 11th Ave. Vancouver, B.C., Canada V5N 1Z7

SUPER CPU ... only from LSI 68008



449.95 kit

We're mighty proud of our new processor card. We're giving you an easy, up to 8000 [illegible] upgrade to your system. Our new CPU gives you these advanced features:

- Dynamic partitioning memory management unit with [illegible]
- On board [illegible] for multi-user/multi-tasking [illegible]
- On board bootstrap EPROM and Monitor EPROM space
- Vectored priority interrupt generator
- On board real time generator
- User selectable bus options that includes a new higher bandwidth bus mode
- And many more

[illegible] CPU ASSEM & TESTED 549.95
[illegible] CPU KIT FORM 449.95
(KIT INCLUDES PROCESSOR, CRYSTAL, SOCKETS AND CONNECTORS)
DISK CONTROLLER SUPPORTED DCJ 20-4, DMF2, 8DC4

LSI STANDS BEHIND ITS PRODUCTS
1 YEAR LIMITED WARRANTY ON ALL OUR PRODUCTS

256K RAM CARD
Using the latest LSI [illegible] this 256K RAM CARD makes a perfect [illegible] to your 56K system. User [illegible] for upward [illegible]
[illegible] 379.95

Announcing . . .

THE *SHELL* FOR FLEX 9™ *

We are pleased to announce the *SHELL*, a UNIX ** like shell that supports I/O redirection, pipes, macro substitution and programmable shell scripts! The shell will work with all your existing programs and utilities. Requires 56K of user ram, FLEX 9™ version 2.0 and above. The shell occupies the top 8K of user ram. An excellent tool for the 6809 community.

FLX/SHO9-8 8 inch version	90.00
FLX/SHO9-5 5.25 inch version	90.00
ONE YEAR MAINTENANCE	22.50

N.Y. residents add sales tax

VISA master charge

and COD accepted

[illegible] is a trademark of Technical Systems Consulting
[illegible] is a registered trademark of Digital Research Inc.
** UNIX is a registered trademark of Bell Labs.
All prices and offers subject to change without notice.

LSI Enterprises Ltd.
PO Box 1227
Woodhaven, NY 11421
(212) 423-5596

DATA ACQUISITION CP/M 2.2

NOW THERE IS AVAILABLE ON THE SS50 BUS INDUSTRIAL QUALITY BOARDS FOR YOUR DEMANDING DATA ACQUISITION NEEDS

ADC1200

- HIGH SPEED 12 BIT A/D BOARD
- 16 CHANNELS Single-ended or Eight Differential
- 25 uSEC CONVERSION TIME
- 80k SAMPLES PER SECOND in single Channel Burst Mode
- INSTRUMENTATION AMPLIFIER / Resistor Selectable Gain
- Contained on Single 30 Pin Board
- CONFIGUREABLE in a Variety of Computer Controlled Modes
- SOFTWARE EXAMPLES
- $795 Each $676.2 to 4

DAC 1220

- HIGH SPEED 12 BIT D/A BOARD
- TWO INDEPENDENT DIGITAL TO ANALOG CONVERTERS
- 10 uSEC SETTLING TIME
- DOUBLE BUFFERED
- BLANKING OUTPUT PULSE
- FOUR QUANDRANT MULTIPLY using EXTERNAL REFERENCE Input
- Contained on Single 30 Pin Board
- $395 Each $336.2 to 4

GPIB4800

- IEEE 488 CONTROLLER BOARD
- Talker, Listener, Controller, Master
- Uses the TI 9914 Controller Chip
- IEEE 488 Panel Mount Connector
- Contained on Single 30 pin Board
- $295 Each 1 to 4

Z809

- CP/M 2.2 OPERATING SYSTEM
- Z80 CO-PROCESSOR
- ASSEMBLER DEBUGGER UTILITIES
- PUBLIC DOMAIN SOFTWARE
- $595 Each $476.2 to 4

WRITE OR CALL TODAY FOR COMPLETE DATA

FOREIGN DEALERS
Kuna Computers • D-5521-Irrel, West Germany
Tel: 652-5299
Digicomp AG • Zurich, Switzerland
Tel: 1-461-12-13
Bernstein Computer Consultants • Cape Town, South Africa • Tel: 21-6394

META LAB

(303) 449-1711
6825 COUNTY LINE ROAD
LONGMONT, CO 80501

TEN MOST-ASKED QUESTIONS about DYNACALC™

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

1. **What is an electronic spread-sheet, anyway?**
   Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

2. **Is DYNACALC just for accountants, then?**
   Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. You can build worksheets as large as 256 columns or 256 rows. There's even a built-in sort command, so you can use DYNACALC to manage small data bases — up to 256 records.

3. **What will DYNACALC do for ME?**
   That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc™, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications. Typical uses include financial planning and budgeting, sales records, bills of material, depreciation schedules, student grade records, job costing, income tax preparation, checkbook balancing, parts inventories, and payroll. But there is no limit to what YOU can do with DYNACALC.

4. **Do I have to learn computer programming?**
   NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Even experienced programmers can get jobs done many times faster with DYNACALC, compared to conventional programming. Built-in HELP messages are provided for quick reference to operating instructions.

5. **Do I have to modify my system to use DYNACALC?**
   Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

6. **Will DYNACALC read my existing data files?**
   You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, STYLOGRAPH™ word processor, RMS™ data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

7. **How fast is DYNACALC?**
   Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

8. **Is there a version of DYNACALC for MY system?**
   Probably. You need a 6809 computer (32k minimum) with FLEX™, UniFLEX™, or OS-9™ operating system. You also need a decent crt terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UniFLEX and OS-9 versions of DYNACALC allow you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX (Frank Hogg or Data-Comp versions).

9. **How much does DYNACALC cost?**
   The FLEX versions are just $200 per copy; UniFLEX version $395; OS-9 version (works with LEVEL ONE or LEVEL TWO) $250. Orders outside North America add $7 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.

10. **Where do I order DYNACALC?**
    See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX or OS-9 versions. Software serial number is required for the UniFLEX version.

Order your DYNACALC today!

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center
13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020

CSC

UniFLEX software prices include maintenance for the first year.

DYNACALC is a trademark of Computer Systems Center

VisiCalc is a trademark of VisiCorp.
STYLOGRAPH is a trademark of Great Plains Computer Co.
RMS is a trademark of Washington Computer Services.
FLEX and UniFLEX are trademarks of TSC.
OS-9 is a trademark of Microware and Motorola.

DYNAMITE+™

"THE CODE BUSTER"

disassembles any 6809 or 6800
machine code program into beautiful source

- Learn to program like the experts!
- Adapt existing programs to your needs!
- Convert your 6800 programs to 6809!
- Automatic LABEL generation.
- Allows specifying FCB's, FCC's, FDB's, etc.
- Constants input from DISK or CONSOLE.
- Automatically uses system variable NAMES.
- Output to console, printer, or disk file.
- Available for all popular 6809 operating systems.

FLEX™ $100 per copy; specify 5" or 8" diskette.
OS-9™ $150 per copy; specify 5" or 8" diskette.
UniFLEX™ $300 per copy; 8" diskette only.

For a free sample disassembly that'll convince you DYNAMITE + is the world's best disassembler, send us your name, address, and the name of your operating system.

Order your DYNAMITE+ today!

See your local DYNAMITE + dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Orders outside North America add $5 per copy. Please specify diskette size for FLEX or OS-9 versions.

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center
13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020

CSC

UniFLEX software prices include maintenance for the first year.

DYNAMITE + is a trademark of Computer Systems Center.

FLEX and UniFLEX are trademarks of TSC.
OS-9 is a trademark of Microware and Motorola.

Dealer inquiries welcome.

'68' MICRO JOURNAL

★ The only ALL 6800 Computer Magazine.

★ **More** 6800 material than all the others combined:

MAGAZINE COMPARISON
(2 years)
Monthly Averages

	6800 Articles			TOTAL
KB	**BYTE**	**CC**	**DOBB'S**	**PAGES**
7.8	6.4	2.7	2.2	19.1 ea. mo.

Average cost for all four each month: **$6.53**
(Based on advertised 1-year subscription price)

'68' cost per month: **$2.04**

That's Right! Much, Much More

for About

1/3 the Cost!

OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card # _____ Exp. Date _____

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: $_____

Name _____

Street _____

City _____ State _____ Zip _____

My Computer Is: _____

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

SUBSCRIPTION RATES

USA
1 Year $24.50, 2 Year $42.50, 3 Year $64.50

*FOREIGN SURFACE Add $12.00 per Year to USA Price

*FOREIGN AIRMAIL Add $36.00 per Year to USA Price

**CANADA & MEXICO Add $5.50 per Year to USA Price
Cash (USA) or drawn on a USA Bank!!!



WINDRUSH MICRO SYSTEMS

UPROM II



PROGRAMS and VERIFIES: I2758, T2508, I2716, T2516, I2732/2732A, MCR68764/8, I2764/2764A, T2564, I27128/27128A, and I27256. I=Intel, T=Texas, M=Motorola.

NO PERSONALITY MODULES REQUIRED!

TRI-VOLT EPROMS ARE NOT SUPPORTED

INTEL's Intelligent programming (tm) implemented for Intel 2764, 27128 and 27256 devices. Intelligent Programming reduces the average programming time of a 2764 from 7 minutes to 1 minute 15 seconds (under FLEX) with greatly improved reliability.

Fully enclosed pod with 5' of flat ribbon cable for connection to the host computer MC6821 PIA interface board.

MC6809 software for FLEX and OS9 (Level 1 or 2, Version 1.2).

BINARY DISK FILE offset loader supplied with FLEX, MDOS and OS9.

Menu driven software provides the following facilities:

a. FILL ..... a selected area of the buffer with a HEX char.
b. MOVE ..... blocks of data.
c. DUMP ..... the buffer in HEX and ASCII.
d. FIND ..... a string of bytes in the buffer.
e. EXAMINE/CHANGE . the contents of the buffer.
f. CRC ..... checksum a selected area of the buffer.
g. COPY ..... a selected area of an EPROM into the buffer.
h. VERIFY ..... a selected area of an EPROM against the buffer.
i. PROGRAM ..... a selected area of an EPROM with data in the buffer.
j. SELECT ..... a new EPROM type (return to types menu).
k. ENTER ..... the system monitor.
l RETURN ..... to the operating system.
l. EXECUTE ..... any DOS utility (only in FLEX and OS9 versions).

FLEX AND OS9 VERSIONS AVAILABLE FROM GIMIX. SSB/MDOS CONTACT US DIRECT.

PL/9

- Friendly interactive environment where you have INSTANT access to the Editor, the Compiler, and the Trace-DeBugger, which, amongst other things, can single step the program a SOURCE line at a time. You also have direct access to any FLEX utility and your system monitor.
- 375+ page manual organized as a tutorial with plenty of examples.
- Fast SINGLE PASS compiler produces 8K of COMPACT and FAST 6809 machine code output per minute with no run-time overheads or license fees.
- Fully compatible with TSC text editor format disk files.
- Signed and unsigned BYTES and INTEGERS, 32-bit floating point REALs.
- Vectors (single dimension arrays) and pointers are supported.
- Mathematical expressions: (+), (-), (*), (/), modulus (%), negation (-)
- Expression evaluators: (=), (<>), (<), (>), (>=), (<=)
- Bit operators: (AND), (OR), (EOR/XOR), (NOT), (SHIFT), (SWAP)
- Logical operators: (.AND), (.OR), (.EOR/XOR)
- Control statements: IF..THEN..ELSE, IF..CASE1..CASE2..ELSE, BEGIN..END, WHILE... REPEAT..UNTIL, REPEAT..FOREVER, CALL, JUMP, RETURN, BREAK, GOTO.
- Direct access to (ACCA), (ACCB), (ACCD), (XREG), (CCR) and (STACK).
- FULLY supports the MC6809 RESET, NMI, FIRQ, IRQ, SWI, SWI2, and SWI3 vectors. Writing a self-starting (free power-up) program that uses ANY, or ALL, of the MC6809 interrupts is an absolute snap!
- Machine code may be embedded in the program via the 'GEN' statement. This enables you to code critical routines in assembly language and embed them in the PL/9 program (see 'MACE' for details).
- Procedures may be passed and may return variables. This makes them functions which behave as though they were an integral part of PL/9.
- Several fully documented library procedure modules are supplied: IOSUBS, BITIO, MATHIO, HEXIO, FLEXIO, SCIPACK, STRSUBS, BASTRING, and REALCON.

'... THIS IS THE MOST EFFICIENT COMPILER I HAVE FOUND TO DATE.'

Quoted from Ron Andersons FLEX User Notes column in '68. Need we say more?

MACE/XMACE/ASM05

All of these products feature a highly productive environment where the editor and the assembler reside in memory together. Gone are the days of tedius disk load and save operations while you are debugging your code.

- Friendly interactive environment where you have instant access to the Editor and the Assembler, FLEX utilities and your system monitor.
- MACE can also produce ASMPROCs (GEN statements) for PL/9 with the assembly language source passed to the output as comments.
- XMACE is a cross assembler for the 6800/1/2/3/8 and supports the extended mnemonics of the 6303.
- ASM05 is a cross assembler for the 6805.

D-BUG

LOOKING for a single step tracer and mini in-line disassembler that is easy to use?? Look no further, you have found it. This package is ideal for those small assembly language program debugging sessions. D-BUG occupies less than 8K (including its stack and variables) and may be loaded anywhere in memory. All you do is LOAD IT, AIM IT and GO! (80 col VDUs only).

McCOSH 'C'

This is as complete a 'C' compiler as you will find on any operating system for the 6809. It is completely compatible with UNIX V7 and only lacks 'Bit-fields' (which are of little practical use in an 8-bit world!).

- Produces very efficent assembly language source output with the 'C' source optionally interleaved as comments.
- Built-in optimizer will shorten object code by about 11%.
- Supports interleaved assembly language programs.
- INCLUDES its own assembler. The TSC relocating assembler is only required if you want to generate your own libraries.
- The pre-processor, compiler, optimizer, assembler and loader all run independently or under the 'CC' executive. 'CC' makes compiling a program to executable object as simple as typing in 'CC,HELLO.C <RETURN>'.

IEEE-488

- SUPPORTS ALL PRINCIPAL MODES OF THE IEEE-488 (1975/8) BUS SPECIFICATION:
  - Talker
  - Listener
  - System Controller
  - Serial Poll
  - Parallel Poll
  - Group Trigger
  - Single or Dual Primary Address
  - Secondary Address
  - talk only ... Listen only
- Fully documented with a complete reprint of the KILOBAUD article on the IEEE bus and the Motorola publication 'Getting aboard the IEEE Bus'.
- Low level assembly language drivers suitable for 6800, 6801, 6802, 6803, 6808 and 6809 are supplied in the form of listings. A complete back to back test program is also supplied in the form of a listing. These drivers have been extensively tested and are GUARANTEED to work.
- Single S-30 board to 8 or 16 addresses per port, fully socketed, gold plated bus connectors and IEEE interface cable assembly.

PRICES

D-BUG	(6809 FLEX only)	$ 75.00
MACE	(6809 FLEX only)	$ 75.00
XMACE	(6809 FLEX only)	$ 98.00
ASM05	(6809 FLEX only)	$ 98.00
PL/9	(6809 FLEX only)	$198.00
'C'	(6809 FLEX only)	$295.00
IEEE-488	with IEEE-488 cable assembly	$298.00
UPROM-II/U	with one version of software (no cable or interface)	$395.00
UPROM-II/C	as above but complete with cable and S-30 interface	$545.00
CABLE	5' twist-n-flat 50 way cable with IDC connectors	$ 35.00
S-30 INT	S-30 interface for UPROM-II	$130.00
EXOR INT	Motorola EXORbus (EXORciser) interface for UPROM-II	$195.00
UPROM SFT	Software drivers for 2nd operating system. Specify FLEX or OS9 AND disk size!	$ 35.00
UPROM SRC	Assembly language source (contact us direct)	

ALL PRICES INCLUDE AIR MAIL POSTAGE

Terms: CWO. Payment by Int'l Money Order, VISA or MASTER-CARD also accepted.

WORSTEAD LABORATORIES, NORTH WALSHAM, NORFOLK, ENGLAND. NR28 9SA.

TEL: 44 (692) 404086
TLX: 975546 WMICRO G

WE STOCK THE FOLLOWING COMPANIES PRODUCTS: GIMIX, SSB, FHL, MICROWARE, TSC, LUCIDATA, LLOYD I/O, & ALFORD & ASSOCIATES.

FLEX (tm) is a trademark of Technical Systems Consultants, OS-9 (tm) is a trademark of Microware Systems Corporation, MDOS (tm) and EXORciser (tm) are trademarks of Motorola Incorporated.

FEATURES THE POWERFUL, THIRD GENERATION, MOTOROLA 6809 PROCESSOR!

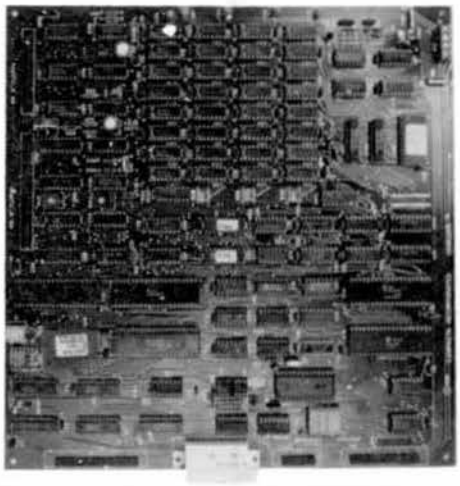
THE 6809 "UNIBOARD"™

SINGLE BOARD COMPUTER KIT

New!

PERFECT FOR COLLEGES, OEM'S, INDUSTRIAL AND SCIENTIFIC USES!

64K RAM! DOUBLE DENSITY FLOPPY DISK CONTROLLER!



BLANK PC BOARD
$99.95
WITH PAL'S, AND TWO EPROMS.

FOR 5-1/4 OR 8 INCH SOURCE DISKETTE ADD $10.

$329.00
COMPLETE KIT! FULLY SOCKETED.

ALL OPTIONS ARE STANDARD. NO EXTRAS TO BUY!

THE COMPACTA UNIBOARD™: Through special arrangement with COMPACTA INC., we are proud to have been selected the exclusive U.S. Mfg. of their new 6809 UNIBOARD™ COMPUTER KIT. Many software professionals feel that the 6809 features probably the most powerful instruction set available today on ANY 8 bit micro. Now, at last, all of that immense computing power is available at a truly unbelievably low price.

FEATURES:

- ★ 64K RAM using 4116 RAMS.
- ★ 6809E Motorola CPU.
- ★ Double Density Floppy Disk Controller for either 5-1/4 or 8 inch drives. Uses WD1793.
- ★ On board 80 x 24 video for a low cost console. Uses 2716 Char. Gen. Programmable Formats. Uses 6845 CRT Controller.
- ★ ASCII keyboard parallel input interface. (6522)
- ★ Serial I/O (6551) for RS232C or 20 MA loop.
- ★ Centronics compatible parallel printer interface. (6522)
- ★ Buss expansion interface with DMA channel. (6844)
- ★ Dual timer for real time clock application.
- ★ Powerful on board system monitor (2732). Features commands such as Go To, Alter, Fill, Move, Display, or Test Memory. Also Read and Write Sectors. Boot Normal, Unknown, and General Flex™.

YOUR CHOICE OF POPULAR DISK OPERATING SYSTEMS:

FLEX™ from TSC	$149
OS9™ from Microware	$199

Specify 5-1/4 or 8 inch

PC BOARD IS DOUBLE SIDED, PLATED THRU SOLDER MASKED, 11 x 11-1/2 IN.

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A FREE COPY IS AVAILABLE UPON REQUEST.

Digital Research Computers
(OF TEXAS)

P.O. BOX 481565 • GARLAND, TEXAS 75046 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. Add $4.00 shipping. USA AND CANADA ONLY

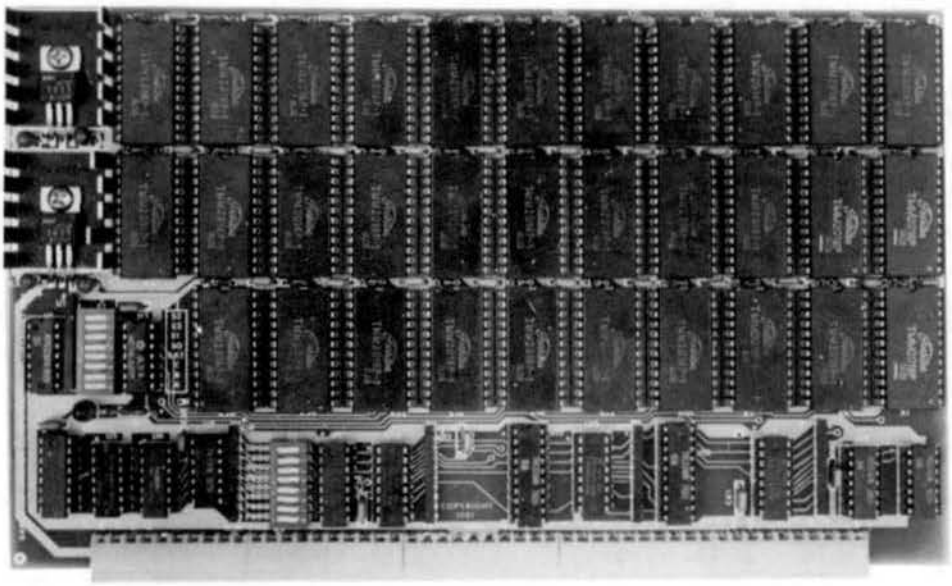
64K SS-50 STATIC RAM

$179.00 (48K KIT)

NEW! NEW!

LOW POWER!

RAM OR EPROM!



BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS -	$18.00
FULL SOCKET SET -	$15.00

ASSEMBLED AND TESTED **ADD $50**

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K	$219
64K	$249

CLOSE OUT SPECIAL
WE HAVE DROPPED OUR 32K SS-50 STATIC RAM BOARD WHICH USED 2114 LOW POWER RAMS. WE WILL SELL THE REMAINING STOCK OF BLANK PCB'S WITH DATA FOR $17.50 EA. THESE FORMERLY SOLD FOR $50.

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Order under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

DISKETTES AND 680X SOFTWARE

SUPER SLEUTH DISASSEMBLER EACH $99-FLEX, $101-OS/9, $100-UNIFLEX
interactively generates source on disk with labels. includes xref, label definition, binary file editing, etc.
specify 6800.1.2.3.5.8.9/6502 version or Z-80/8080/85 version

(OBJECT ONLY) EACH $50-FLEX & OS/9, $49-COCO DOS
COCO DOS available in 6800.1.2.3.5.8.9/6502 version only

CROSS-ASSEMBLERS EACH $50-FLEX, $55-OS/9, $60-UNIFLEX, ALL $100
specify for 6800/1, 6502, 6805, Z-80, or 6800/48/85
OS/9 version requires Microware RMA or Lloyd OSM macro assembler
FLEX version requires TSC ASM6 or FHL ASM or OSM macro assembler

DEBUGGING SIMULATORS EACH $75-FLEX, $100-OS/9, $80-UNIFLEX
specify 6800/1, 6805/146805, 6502, or (6809 OS/9 only)
object only for COCO FLEX and COCO OS/9 users: $50 each

6502 TO 6809 ASSEMBLER TRANSLATOR $75-FLEX, $85-OS/9, $80-UNIFLEX
translates 6502 programs to 6809, noting inexact conversions

6800 TO 6809 & 6809 PIC TRANSLATORS $50-FLEX, $75-OS/9, $60-UNIFLEX
translates 6800 programs to 6809, 6809 programs to PIC

FULL-SCREEN FLEX AND UNIFLEX TSC XBASIC PROGRAMS FOR 6809
(with complete cursor control)

DISPLAY GENERATOR/DOCUMENTOR	$50 w/source, $25 without
MAILING LIST SYSTEM	$100 w/source, $50 without
INVENTORY WITH MRP	$100 w/source, $50 without
TABULA RASA SPREADSHEET	$100 w/source, $50 without

DISK UTILITY PROGRAM LIBRARY $50-FLEX
edit disk sectors, sort directory, maintain master catalog, etc. (requires TSC XBASIC)

CMODEM PROGRAM (OBJECT ONLY) $50-FLEX & OS/9 & UNIFLEX
provides menu-driven telecommunications facilities, with terminal mode, up/down load, MODEM7 protocol, etc.

5.25" SOFT-SECTORED DISKETTES EACH SET OF 10 $14-SSDD, $17-DSDD
with Tyvek jackets, hub rings, labels

Most programs in source on disk: specify computer, disk size, operating system.
Contact CSC for full catalog and dealer information.
25% discount for multiple purchases of same program on same order.
For VISA and MASTER CARD, give account, exp. date, phone. US funds only.
Add 5% shipping; no shipping charge for diskettes in lots of 100.
(UNI)FLEX trademark Technical Systems Consultants. OS/9 trademark Microware.

Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717/4570

SOFTWARE FOR THE HARDCORE

** FORTH PROGRAMMING TOOLS from the 68XX&X **
** FORTH specialists — get the best!! **

NOW AVAILABLE — A variety of rom and disk FORTH systems to run on and/or do TARGET COMPILATION for

6800, 6301/6801, 6809, 68000, 8080, Z80

Write or call for information on a special system to fit your requirement.

Standard systems available for these hardware —

EPSON HX-20 rom system and target compiler
6809 rom systems for SS-50, EXORCISER, STD, ETC.
COLOR COMPUTER
6800/6809 FLEX or EXORCISER disk systems.
68000 rom based systems
68000 CP/M-68K disk systems, MODEL II/12/16

tFORTH is a refined version of FORTH Interest Group standard FORTH, faster than FIG-FORTH. FORTH is both a compiler and an interpreter. It executes orders of magnitudes faster than interpretive BASIC. MORE IMPORTANT, CODE DEVELOPMENT AND TESTING is much, much faster than compiled languages such as PASCAL and C. If Software DEVELOPMENT COSTS are an important concern for you, you need FORTH!

firmFORTH™ is for the programmer who needs to squeeze the most into roms. It is a professional programmer's tool for compact rommable code for controller applications.

™ tFORTH and firmFORTH are trademarks of Talbot Microsystems
™ FLEX is a trademark of Technical Systems Consultants, Inc.
™ CP/M-68K is trademark of Digital Research, Inc.

tFORTH™ from TALBOT MICROSYSTEMS NEW SYSTEMS FOR 6301/6801, 6809, and 68000

---> tFORTH SYSTEMS <---

For all FLEX systems: GIMIX, SWTP, SSB, or EXORcisor Specify 5 or 8 inch diskette, hardware type, and 6800 or 6809.

** tFORTH — extended fig FORTH (1 disk) with fig line editor. $100 ($15)
** tFORTH + — more! (3 5" or 2 8" disks) $250 ($25)
adds screen editor, assembler, extended data types, utilities, games, and debugging aids.
** TRS-80 COLORFORTH — available from The Micro Works
** firm FORTH — 6809 only. $350 ($10)
For target compilatio s to rommable code.
Automatically deletes unused code. Includes HOST system source and target nucleus source. No royalty on targets. Requires but does not include tFORTH +.
** FORTH PROGRAMMING AIDS — elaborate decompiler $150

** tFORTH for HX-20, in 16K roms for expansion unit or replace BASIC $170

** tFORTH/68K for CP/M-68K 8" disk system $290
Makes Model 16 a super software development system.

** Nautilus Systems Cross Compiler
— Requires: tFORTH + HOST + at least one TARGET:
— HOST system code (6809 or 68000) $200
— TARGET source code: 6800-$200, 6301/6801—$200
same plus HX-20 extensions— $300
6809—$300, 8080/Z80—$200, 68000—$350

Manuals available separately — price in ( ).
Add $6 system for shipping, $15 for foreign air.

TALBOT MICROSYSTEMS 1927 Curtis Ave., Redondo Beach, CA 90278 (213) 376 9941

!!! FREE !!!

Published Monthly by Computer Publishing Inc., Hixson, TN.

$1.95

Bulk Rate
U.S. Postage
PAID
Chattanooga, TN
Permit No. 357



Color Micro Journal

The Color Computer Monthly Magazine

$1.95 per issue Vol. 1, Issue 2 October, 1983

THIS 'N THAT

The BIG NEWS this month is that OS-9 has finally arrived for the Color Computer. The ASTOUNDING part of the Radio Shack OS-9 Package, besides the price, is the [illegible]. You 'Old Time Radio Shack Followers' will not believe what you see. Jon Shirley has been telling us that the main reason for the "lack" of documentation with a lot of their products was the restrictions placed on releasing that information by Microsoft; I

OS-9 on the COLOR COMPUTER

One of the "Operating Systems of the Future" is now available for the "little old Color Computer": OS-9. Freely translated, OS-9 means "Operating System for the 6809" (OS-9 is now being written for the 6800, also). Since it is fairly obvious that UNIX and "UNIX-Type" Operating Systems will be running on just about every computer to come out in the next few years, a whole new language is beginning to appear on the horizon.

Color Computer OS-9; the Package

We had been running a preliminary release of OS-9 on the Color Computer for a few weeks, and received the "Official Radio Shack" version for Review a couple of days ago. To put it mildly, this package is IMPRESSIVE! For $69.95 (Radio Shack Catalog Number 26-3020), you receive a 9 1/2" x 7 5/8" x 2" package containing 4

FREE SAMPLE ISSUE

1-800-338 6800

MON.-FRI. 9-5 E.S.T.

TELEX 558 414 PVT BTH

USA-$12.50 per year. Canada& Mexico-$19.50 per year

Surface Foreign-$24.50 per year. Airmail Foreign-$48.50 per year

Color Micro Journal™

TM Color Micro Journal is a trademark of Computer Publishing Inc.

5900 Cassandra Smith Rd. Hixson,TN. 37343

6809 Word Processing System

stylograph™

SPELLING CHECKER
Checks all words against an internal user-expandable dictionary of over 42,000 words.
OS9, FLEX $145 UniFLEX $195

STYLOGRAPH 2.0
The "User Friendly" word processing system. Fewer key strokes by the operator make it easier to learn.
OS9, FLEX $295 UniFLEX $395
COLOR COMPUTER FLEX $195

MAIL MERGE
Inserts names and addresses into form letters and mailing lists. Appends files at print out time. Handles files longer than memory.
OS9, FLEX $125 UniFLEX $175

Inquire about our other software
- Business Programs - G/L, A/R, A/P
- Data Base Management System
- Assemblers

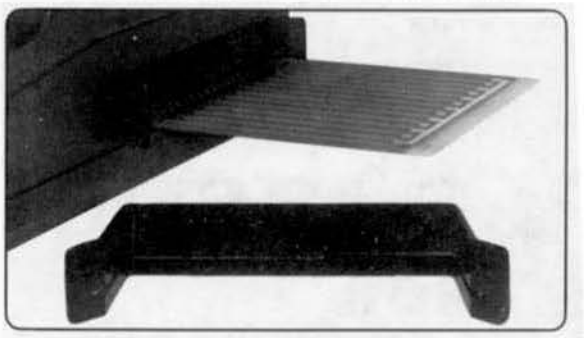
Also, Daisy Wheel Printers $599.

Great Plains Computer Company Inc.
P.O. Box 916
Idaho Falls, Idaho 83401

(208) 529-3210

Flex and Uniflex are trademarks of Technical Systems Consultants, Inc.
OS9 is a trademark of Microware.

6809 SYSTEM DEVELOPMENT



EXPANSION HARDWARE FOR THE TRS-80 COLOR COMPUTER

XPNDR1™

CoCo Expander Card

Gold edge connector plugs into the CoCo cartridge connector. Signals are labeled on the bottom (wire side) with ground and power buses; plated through holes. The 4.3 × 6.2 inch glass/epoxy card is drilled for ICs and components. The finest bare breadboard for your CoCo. Includes 8 page *Application Notes* to help you get started.

$19.95 each or 2 for $36

SuperGuide™

Precision molded plastic insert designed specifically to align and support printed circuit cards in the CoCo cartridge slot; an unbreakable removable card guide. Patent Pending.

$3.95 each

Available now from:

ROBOTIC MICROSYSTEMS

BOX 30807 SEATTLE, WA 98103

K-BASIC
for OS9 & FLEX
$199

K-BASIC is a complete BASIC compiler package including: the compiler itself; the assembler; documentation; and sample programs. It features six atomic data types including: real numbers; strings; 8 bit, 16 bit, 32 bit, and 64 bit signed integers. All types may be dimensioned with one or two subscripts. K-BASIC converts programs to MACHINE language code which may be put into EPROMS or ROMS.

K-BASIC syntax is very close to TSC's BASIC and X-BASIC interpreters. Line numbers are not required (may be up to 16 characters). Variable names may be up to 12 characters long. The AT statement dimensions variables to absolute memory addresses.

The future of K-BASIC will see additional versions for the assorted interpreters currently available. This means you can compile your BASIC programs you now have.

Call (503) 666-1097 for our CATALOG, we have many other programs including: DO ... $69 OSM ... $99 ED/ASM ... $69

CRASMB
for OS9 & FLEX
$399

CRASMB is the highly acclaimed cross assembler package for OS9 and FLEX systems, and is the only one of its type available. It turns your computer into a development station for these CPUs:

6800 6801 6804 6805 6809 6811 6502
7000 1802 8048 8051 8080 8085 Z80
(68000 16/32 bit cross assembler ... $249)

CRASMB features include: Macros, Conditional assembly, Library file calls (12 deep), Symbol length to 30 characters, Symbol cross reference tables, Object code in 4 formats (OS9, FLEX, S1-S9, INTEL HEX), plus many other extended directives and options not found on other assemblers.

LLOYD I/O 19535 NE GUSAN, PORTLAND, OR 97230 USA
Phone: (503) 666-1097 (Software Consultation Available)

VISA, MC, COD, CHECK, APPROVED P.O.'s ACCEPTED

England: Vivaway (0582 423425), Windrush (0692 405189)
Germany: Zacher Computer (65 25 299)
Australia: Paris Radio Electronics (61 2 344 9111)

OS9 is a ™ of Microware. FLEX is a ™ of TSC

OS-9®
SOFTWARE TOOLS!

- **ADLIB 50.00**
  Allows shared source code
- **ISAM 350.00**
  Indexed Sequential File Access Method
- **SMATH 95.00**
  String Arithmetic
- **SORTC 150.00**
  Full-record disk sort
- **EXAMOD/CHGREV 50.00**
  Determines contents of a module
- **IN-DATA/REPORT-GEN/MENU 75.00**
  Data entry/Menu & Report generator
- **LOOKUP/SLOOKUP 75.00**
  "Wild Card" directory searcher
- **VID 75.00**
  Display/input data edit package
- **XRF 200.00**
  Low-overhead database emulator
- **TERMINAL 95.00**
  Communicates with other machines
- **LDMAC 75.00**
  Assembly code routines "tool box"

For more information, or to place an order, contact:
Dept. 68 8
The JBM Group, Inc.
Continental Business Center
Front & Ford Streets
Bridgeport, PA 19405
Tel: 215-337-3138/TWX: 510-660-3999

the JBM group

VISA/MC accepted: PA residents please add 6% sales tax
U.S. orders please add 5.00 postage/handling

* Registered trademark of Microware

STAR-DOS LEVEL I

Whenever a new DOS is introduced, there's always the problem of developing software to work with it. So we did it the opposite way — we analyzed the requirements of software that already exists and developed a DOS that met them... and exceeded them! The result is STAR-DOS Level I, a new DOS for 6809 systems, ideal for single-user industrial, control, and advanced hobbyist applications. This includes SS-50 systems and single-board computers from a variety of vendors.

Level I is compatible with most current 6809 hardware and software. On the hardware side, it allows up to ten floppy or Winchester drives with appropriate controllers. On the software side, it runs existing 6809 software from all the major 6809 software suppliers, including TSC, Star-Kits, Introl, and others.

Write or call for more information. STAR-KITS Software Systems Corporation. P.O. Box 209, Mt. Kisco N.Y. 10549 (914) 241-0287.

STAR-KITS

ANDERSON COMPUTER CONSULTANTS
&
Associates

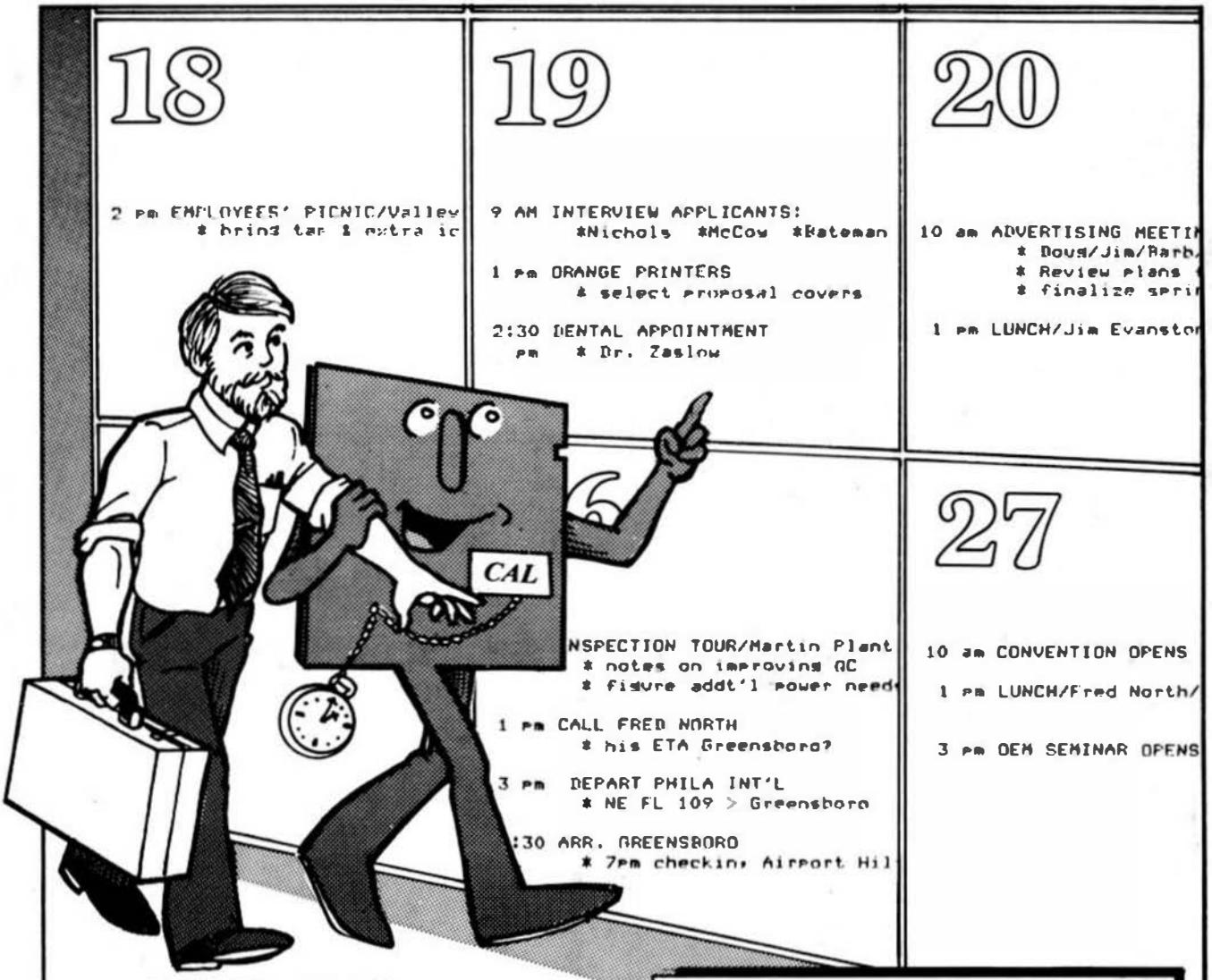
Ron Anderson, respected author and columnist for 68 MICRO JOURNAL announces the **Anderson Computer Consultants & Associates,** a consulting firm dealing primarily in 68XX(X) software design. Our wide experience in designing 6809 based control systems for machine tools is now available on a consultation basis.

Our experience includes programming machine control functions, signal analysis, multi-axis servo control (CNC) and general software design and development. We have extensive experience in instrumentation and analysis of specialized software. We support all popular languages pertaining to the 6809 and other 68XX(X) processors.

If you are a manufacturer of a control or measuring package that you believe could benefit from efficient software, write or call Ron Anderson. The fact that any calculation you can do with pencil and paper, can be done much better with a microcomputer. We will be happy to review your problem and offer a modern, state-of-the-art microcomputer solution. We can do the entire job or work with your software or hardware engineers.

Anderson Computer Consultants & Associates
3540 Sturbridge Court
Ann Arbor, MI 48105

WHO? WHAT? WHEN? WHERE?
Let CAL Help You Prepare...



JBM's CAL program ($69) is the fastest, simplest, most comprehensive schedule manager available for OS9* (all levels).

MasterCard

VISA

For more information or to place an order, contact:

Dept. 68 13

The JBM Group, Inc.
Continental Business Center
Front & Ford Streets
Bridgeport, PA USA 19405
TEL: 215-337-3138
TWX: 510-660-3999

the JBM group

VISA/MASTERCHARGE accepted. PA res. add 6% sales tax.
US orders, add $5.00 postage and handling.

* OS9 is a registered trademark of Microware Corp.

ARCADE 50

POWERFUL COLOR GRAPHICS
Uses the new TMS9918A Video Display procesor. High resolution 256 x 192 pixel display with 15 colors. 16K Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation. External Video input allows subtitling. NTSC composite video output.

SOUND EFFECTS AND MUSIC
- Three AY3-8910 Programmable Sound Generators
- Nine simultaneous voices
- Three independent noise sources
- Onboard stereo amplifier drives two 8 ohm speakers

ADDITIONAL I/O CAPABILITIES
- Eight analog inputs with 8 bit resolution
- Supports four joysticks with pushbutton switches
- Eight bit parallel I/O port
- Entire unit maps into 256 bytes of memory

FBASIC

TERMINUS DESIGN INC. in conjunction with Microware Systems Corporation is proud to announce FBASIC an enhancement of Microware's 6800/BASIC. Their fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions, Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50.

ARCADE 50 assembled and tested	$325.00
Video and Audio connector set	15.00
4 Joystick connector set	15.00
2 Radio Shack Joysticks	24.00
Gold Molex connectors	12.00
A/BASIC for 6	110.00
FBASIC for 6809	110.00
FBASIC (with ARCADE 50)	75.00
ARCADE 50 RGB	375.00
LABVIDEO (Motorola EXORbus)	375.00
NEW MV09 6809 Processor Board	225.00
256K Dynamic Memory Board	795.00
256K Dynamic Memory Board (w/64K)	395.00
64K Dynamic Memory Board	295.00

TERMINUS DESIGN INC
16 SCARBROUGH ROAD
ELLENWOOD, GA 30049
(404) 474-4866

OS9
APPLICATION SOFTWARE

ACCOUNTS PAYABLE **$349**

GENERAL LEDGER with CASH JOURNAL **$449**

PAYROLL **$549**

ACCOUNTS RECEIVABLE **$349**

SMALL BUSINESS INVENTORY **$349**

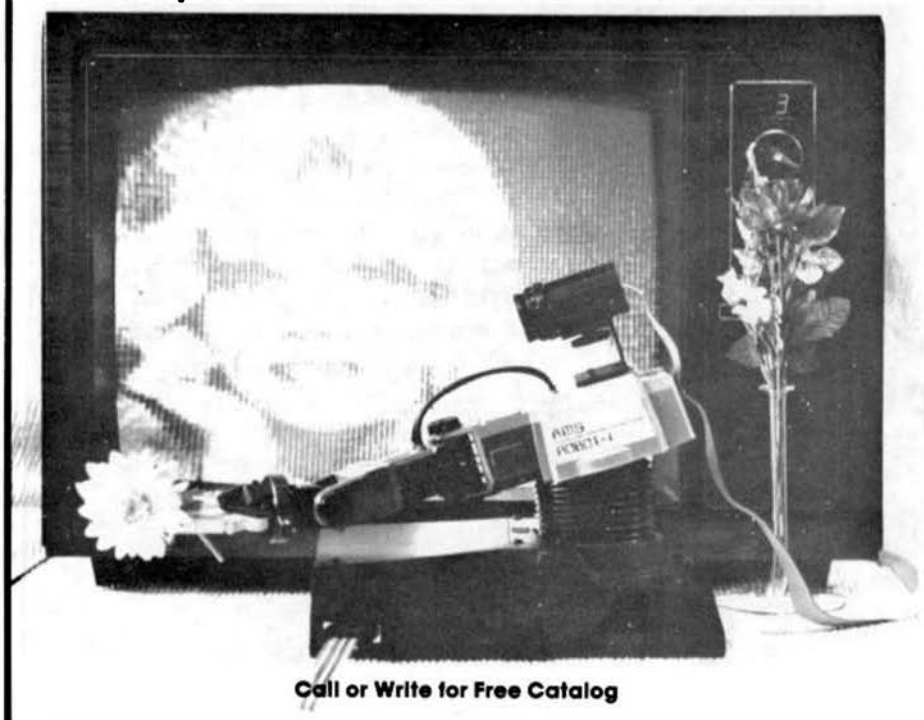
COMPLETE DOCUMENTATION $19.95

OS9 & BASIC 09 ARE TRADEMARK OF MICROWARE, INC. & MOTOROLA CORP.

SPECIALTY ELECTRONICS

(405) 233-5564
2110 W. WILLOW — ENID, OK 73701

Computer Servo Controlled Robot Arm



Call or Write for Free Catalog

Analog Micro Systems

5660 Valmont Road · Boulder, Colorado 80301 · Tel: (303) 444-6809

Robot-1

Keyboard or Joystick Control

Remembers Everything It Did & does it again

Typical System Includes:
- Robot-1 & Cables
- 6 Channel Servo Controller
- Power Supply
- All Software with source code

Modular Robotic Accessories:
- Mobile Cart for Traveling Robot
- Radio Links between all Functions
- Robot-mounted MicronEye
- Ultrasonic Range Finder

Robot-1 Series

starting at $289.00
for the Color Computer
and 6809 SS50 Computers

Additional Systems Available

Robot-1R for Radio Control Systems

ACORN

COMPUTER SYSTEMS SS-50C



MODULES - BARE CARDS - KITS - ASSEMBLED & TESTED

Stackable Modules		KIT	A&T
20 amp POWER SUPPLY w/fan w/Disk protect relay		350.00	400.00
DISK CABINET w/rebs. & cables less DRIVES		200.00	250.00
MOTHER BOARD, 8 SS-50c, 8 SS-30c NMI button		225.00	325.00
Item	**Bare**	**KIT**	**A&T**
IT3 - INTERRUPT TIMER 1, 10, 100 per sec.	19.95	29.95	39.95
PB4 - INTELLIGENT PORT BUFFER Single board comput.	39.95	114.95	139.95
DP1A - Dual PIA p rallel port, 4 buffered I/Os	24.95	69.95	89.95
XADR - Extended Addressing BAUD gen. PIA port	29.95	69.95	89.95
MB8 - MOTHER BOARD SS-50c w/BAUD gen.	64.95	149.95	199.95
P16K - 16K PROM DISK 21, 2764 EPROMs	39.95	79.95	109.95
FD69 - Firmware development 2, 8K blocks	39.95	84.95	114.95
EXPR - 2764 PROM burner adapt. for 2716 BURNER		19.95	------
CHERRY Keyboard w/Cabinet 96 key capacitive		249.95	------
TAXAN 12", 18 Mhz MONITOR GREEN		------	149.95
AMBER		------	159.95
4 MODULE CABINET - unfinished		150.00	------
POWER SUPPLY w/disk protect		250.00	------

Color Computer

MODOLINK - 20 Mhz Monochrome video driver		15.00	20.00
CC3O PORT BUS w/power supply 5 SS-30, 2 C rt		169.95	199.95
POWER BOX 6 switched outlets transient suppression		29.95	39.95
RS-232 3-switched ports for above	ADD	+20.00	+25.00

Write for FREE Catalog

ADD $3.00 S&H PER ORDER

WIS. ADD 5% SALES TAX

11931 W. Bluemound Road

MILWAUKEE. WIS. 53226

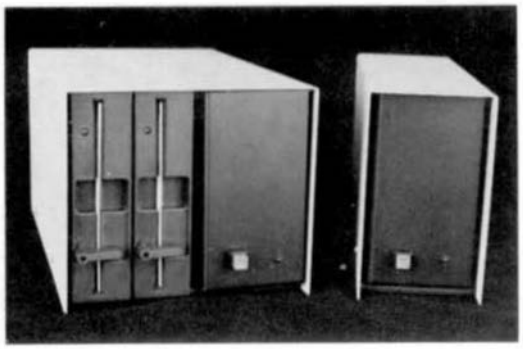
(414) 257 - 0300

68' MICRO JOURNAL ADVERTISERS INDEX

'68' MICRO JOURNAL 48,60
AAA CHICAGO COMPUTER CENTER 36,37
ACORN COMPUTER SYSTEMS 70
ANALOG MICRO SYSTEMS 69
ANDERSON COMPUTER CONSULTANTS 67
COLOR MICRO JOURNAL 65
COMPILER EVALUATION SERVICES 48
COMPUTER PUBLISHING INC. 5
COMPUTER SYSTEMS CENTER 59,60
COMPUTER SYSTEMS CONSULTANTS, INC. ...64
DATA-COMP .....IBC
DIGITAL RESEARCH COMPUTERS 62,63
GIMIX, INC. 3,72
GREAT PLAINS COMPUTER CO. 66
HAZELWOOD COMPUTER SYSTEMS .....OBC
INTROL CORP. 50
JBM 67,68
LLOYD I/O 66
LSI ENTERPRISES LTD. 58
META LAB 58
MICROWARE SYSTEMS CORP. 1,4
OMEGASOFT CERTIFIED SOFTWARE CORP. ...51
PERIPHERAL TECHNOLOGY 71
RACAL-DANA INSTRUEMENTS 49
ROBOTIC MICROSYSTEMS 66
SARDIS TECHNOLOGIES 58
SMOKE SIGNAL BROADCASTING 6,7
SOUTH EAST MEDIA 52,53,54,55,56,57
SOUTHWEST TECHNICAL PRODUCTS INC. ...IFC
SPECIALTY ELECTRONICS 69
STAR-KITS 67
TALBOT MICROSYSTEMS 64
TERMINUS DESIGN, INC. 69
UNITEK 58
WESTCHESTER APPLIED BUSINESS SYSTEMS .71
WINDRUSH MICRO SYSTEMS LIMITED 61

This index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

PT69 SINGLE BOARD COMPUTER SYSTEM
OS-9 NOW AVAILABLE



Pictured:
System with Drives/System without Drives

The proven PT69 Single Board Computer now features OS-9 capability. Powerful performance, reliability, + OS-9 — UNBEATABLE! The PT69 is a complete system in a compact package.

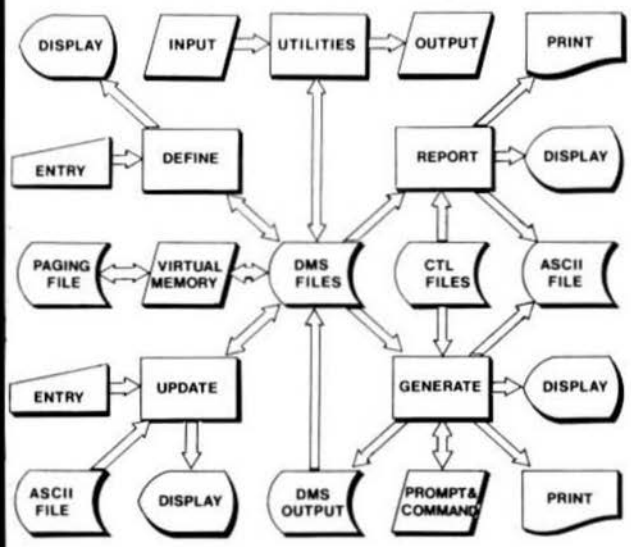
- 1 MHZ 6809E Processor
- 2 RS232 Serial Ports (6850)
- 2 8-Bit Parallel Ports (6821)
- 56K RAM; 4K EPROM
- Time-of-Day Clock (MC146818)

* COMPLETE SYSTEM with PT69 Board, 2 DS/DD 5¼" 40 Track Drives, Cabinet, and Power Supply	$999.95
* PT 69 Board, Assembled and Tested, with Power Supply + Cabinet	$399.95
* PT69, Assembled and Tested Board	$299.95
* Parallel Printer interface with cables	$ 49.95
* OS-9 L1, includes edit, asm, + debug	$250.00
* STAR-DOS Level 1 (Compatible with Flex)	$ 75.00

PERIPHERAL TECHNOLOGY
"Supplying Your Computer Needs Since 1978"
3670 Lower Roswell Road
Marietta, Georgia 30067
VISA/MASTERCARD/CHECK/COD 404/973-0042

™OS-9 is a trademark of Microware and Motorola
™FLEX is a trademark of Technical Systems Consultants

XDMS
Data Management System



System Architecture

WESTCHESTER Applied Business Systems
Post Office Box 187
Briarcliff Manor, N.Y. 10510

XDMS Data Management System
The XDMS Data Management System is available in three levels. Each level includes the XDMS nucleus, VMGEN utility and System Documentation for level III. XDMS is one of the most powerful systems available for 6800 computers and may be used for a wide variety of applications. XDMS users are registered in our database to permit distribution of product announcements and validation of user upgrades and maintenance requests.

XDMS Level I
XDMS Level I consists of DEFINE, UPDATE and REPORT facilities. This level is intended as an "entry level" system, and permits entry and reporting of data on a "tabular" basis. The REPORT facility supports record and field selection, field merge, sorting, line calculations, column totals and report titling. Control is via a English-like language which is upward compatible with level II. XDMS Level I $129.95

XDMS Level II
Level II adds to Level I the powerful GENERATE facility. This facility can be thought of as a general file processor which can produce reports, forms and form letters as well as file output which may be re-input to the facility. GENERATE may be used in complex processing applications and is controlled by a English-like command language which encompasses that used by Level I. XDMS Level II $195.95

XDMS Level III
Level III includes all of level II plus a set of useful DMS Utilities. These utilities are designed to aid in the development and maintenance of user applications and permit modification of XDMS system parameters, input and output of XDMS files, display and modification of file format, graphic display of numerical data and other functions. Level III is intended for advanced XDMS users. XDMS Level III $269.95
XDMS System Documentation only $10, credit toward purchase. . . $ 24.95

XACC Accounting System
The XACC General Accounting System is designed for small business environments of up to 10,000 accounts and inventory items. The system integrates accounting functions and inventory plus the general ledger, accounts receivable and payable functions normally sold separately in other systems. Features user defined accounts, products (or services), transactions, invoicing, etc. Easily configured to most environments. XACC General Accounting System (Requires XDMS, prof. Lv. III). . $290.95
XACC System Documentation only $10, credit toward purchase. . . $ 24.95

WESTCHESTER Applied Business Systems
Post Office Box 187, Briarcliff Manor, N.Y. 10510

All software is written in macro/assembler and runs under 6809 FLEX O/S. Terms: Check, Money Order, Visa or Mastercharge. Shipment first class. Add P&H $2.50 ($7.50 Foreign). NY Res add sales tax. Specify 5" or 8".

Sales: S. E. MEDIA, 1-800-338-6800, Consultation: 914-941-3552 (evens).

FLEX is a trademark of Technical Systems Consultants, Inc.

GIMIX HAS THE 6809 SYSTEM TO SUIT YOUR NEEDS

HARDWARE

All systems feature the **GIMIX CLASSY CHASSIS**; with a ferro-resonant constant voltage power supply, gold plated bus connectors, and plenty of capacity for future expansion.

Static **RAM** and double-density **DMA** floppy disk controllers are used exclusively in all systems.

All systems are guaranteed for 2 MHz operation and include complete hardware and software documentation, necessary cables, filler plates, etc.

Systems are assembled using burned-in and tested boards, and all disk drives are tested and aligned by **GIMIX**.

You can add additional components to any system when ordering, or expand it in the future by adding **RAM, I/O**, etc.

GIMIX lets you choose from a wide variety of options to customize your system to your needs.

SOFTWARE

All **OS-9/FLEX** systems allow you to software select either operating system.

Also included is the **GMXBUG** monitor and, in systems with 128K or more of RAM, **GMX-VDISK** for **FLEX**.

All **GIMIX OS-9** systems include **Microware's Editor, Assembler, Debugger, Basic09,** and **Runb**; and the GMX versions of **RMS** and **DD** for **OS-9**.

All **GIMIX** versions of **OS-9** can read and write RS color computer format OS-9 disks, as well as the Microware/GIMIX standard format.

New and exclusive with **OS-9 GMX III** systems is the **GMX OS-9 Support ROM**, a monitor for **OS-9** that includes memory diagnostics and allows the system to boot directly from either hard disk or floppy.

A wide variety of languages and other software is available for use with either **OS-9** or **FLEX**.

OS-9 GMX III/FLEX SYSTEMS (#79)

The **#79** super system now includes (in addition to the above): the **GMX 6809 CPU III, a 256K CMOS Static RAM Board (#72), and a 3-port Intelligent Serial I/O Processor (#11)**.

The **GMX 6809 CPU III** can perform high-speed DMA transfers from memory to memory and uses memory attributes and illegal instruction trapping to protect the system and users from program crashes. If a user program crashes, only that user is affected; other users are unaware of the problem.

The 3-Port Intelligent Serial I/O Board (#11) significantly reduces system overhead by handling routine I/O functions; freeing the host CPU for running user programs. This improves overall system performance and allows user terminals to be run at up to 19.2K baud.

with dual 40 track DSDD drives	**$5998.79**
with dual 80 track DSDD drives	**$6198.79**
with **#88** dual 8" DSDD drive system	$7698.79
with **#90** 19MB Winchester subsystem and one 80 track	**$8898.79**
with a 47MB Winchester subsystem and one 80 track	$10,898.79
with a 47MB plus a 6MB removable pack Winchester subsystem and one 80 track drive	$12,398.79

TO ORDER BY MAIL: SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add $5 handling if order is under $200.00. Foreign orders add $10 handling if order is under $200.00. Foreign orders over $200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, 231 S. LaSalle Street, Chicago, IL 60693, account #73-32033.

BASIC-09 and OS-9 are trademarks of Microware Systems Corp. and MOTOROLA, Inc.
FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.
GIMIX, GHOST, GMX, CLASSY CHASSIS, are trademarks of GIMIX, Inc.

OS-9 GMX I / FLEX SYSTEMS #49

The **#49** systems include 64KB static RAM, #05 CPU, #43 2 port serial board.

with dual 40 track DSDD drives	**$3998.49**
with dual 80 track DSDD drives	**$4198.49**
with **#88** dual 8" DSDD drive system	**$5698.49**
with **#90** 19MB Winchester subsystem and one 80 track	**$6898.49**

OS-9 GMX II / FLEX SYSTEMS #39

The **#39** systems include 128KB static RAM, #05 CPU, #43 2 port serial board.

with dual 40 track DSDD drives	**$4498.39**
with dual 80 track DSDD drives	**$4698.39**
with **#88** dual 8" DSDD drive system	**$6198.39**
with **#90** 19MB Winchester subsystem and one 80 track	**$7398.39**

GIMIX DOES NOT GUARANTEE PERFORMANCE OF ANY GIMIX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS PRODUCT.

EXPORT MODELS: ADD $30 FOR 50Hz. POWER SUPPLIES.

GIMIX, Inc. reserves the right to change pricing, terms, and products specifications at any time without further notice.

ALL PRICES ARE F.O.B. CHICAGO

Contact GIMIX for price and availability of UniFLEX and UniFLEX GMXIII Systems.

NOTE on all drive systems: Dual 40 track drives have about 700KB of formatted capacity; dual 80's about 1.400KB; dual 8" about 2.000KB. The formatted capacity of hard disks is about 80% of the total capacity.

Want to expand your system to a megabyte of Static RAM and 15 users?

Simply add additional memory and I/O boards. Your GIMIX system can grow with your needs. Contact us for a complete list of available boards and options.

#72 256KB CMOS STATIC RAM board with battery back up	**$1898.72**
#64 64KB CMOS STATIC RAM board with battery back up	**$528.64**
#67 64KB STATIC RAM board	**$478.67**
#11 3 port intelligent serial I/O board	**$498.11**
#43 2 port serial I/O board	**$128.43**
#42 2 port parallel I/O board	**$88.42**
#95 cable sets (1 needed per port), specify board	**$24.95**

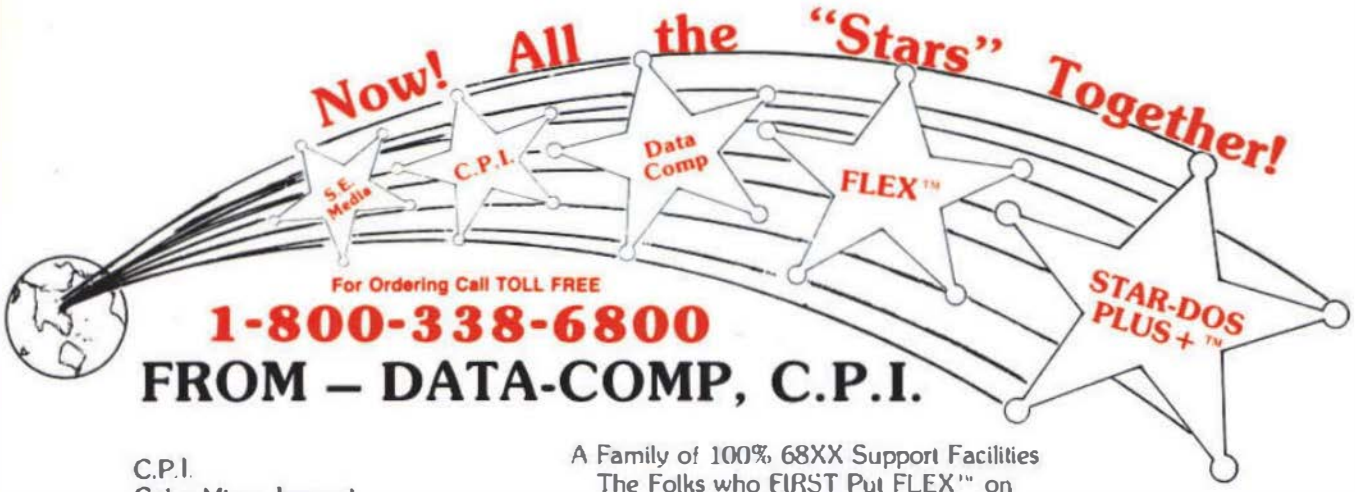
TRADE UP YOUR CoCo!

GIMIX will allow you up to $1100.00 credit toward the purchase of any GIMIX system when you trade-in your working Color Computer, peripherals, and original software. The trade-in value is limited to 110% of the RADIO SHACK™ list price at the time your order is placed. You pay the freight. This offer is good only in the Continental U.S.; is limited to the first 100 orders; and expires on 9/30/84. Only one trade-in per customer.



GIMIX INC.
1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510 • TWX 910-221-4055

©1984 GIMIX, INC. 4-84



For Ordering Call TOLL FREE

1-800-338-6800

FROM – DATA-COMP, C.P.I.

C.P.I.
Color Micro Journal
'68' Micro Journal
Data-Comp
S.E. Media

A Family of 100% 68XX Support Facilities
The Folks who FIRST Put FLEX™ on
The CoCo
Now Offering: *FLEX™ (2 Versions)
AND *STAR-DOS PLUS+™

FLEX-CoCo Sr.
with TSC Editor
TSC Assembler
Complete with Manuals
Reg. $250.00
Only $79.95

STAR-DOS PLUS+
- Functions Same as FLEX
- Reads - writes FLEX Disks
- Run FLEX Programs
- Just type: Run "STAR-DOS"
- Over 300 utilities & programs to choose from.

$34.50

FLEX-CoCo Jr.
without TSC Editor & Assembler
$49.95

PLUS

ALL VERSIONS OF FLEX & STAR-DOS• INCLUDE

TSC Editor
Reg $50.00
NOW $35.00

+ Read-Write-Dir RS Disk
+ Run RS Basic from Both
+ More Free Utlities
+ Super 800 Support
+ Free Color Micro Journal 1 yr. sub.
+ External Terminal Program
+ Test Disk Program
+ Disk Examine & Repair Program
+ Memory Examine Program
+ Many Many More!!!

TSC Assembler
Reg $50.00
NOW $35.00

DISK SYSTEMS FOR THE COLOR COMPUTER

THESE PACKAGES INCLUDE DRIVE, *CONTROLLER, POWER SUPPLY & CABINET, CABLE, AND MANUAL.

* SPECIFY WHAT CONTROLLER YOU WANT J&M, OR RADIO SHACK.

PAK #1 - 1 SINGLE SIDED, DOUBLE DENSITY SYS.	$389.95
PAK #2 - 2 SINGLE SIDED, DOUBLE DENSITY SYS.	$639.95
PAK #3 - 1 DOUBLE SIDED, DOUBLE DENSITY SYS.	$439.95
PAK #4 - 2 DOUBLE SIDED, DOUBLE DENSITY SYS.	$699.99
PAK #5 - 2 DOUBLE SIDED, DOUBLE DENSITY SYS. THINLINE DRIVES, HALF SIZE	$699.99
COLOR COMPUTER II 64K W/EXT. BASIC	$189.95

CONTROLLERS

J&M DISK CONTROLLER W/ DOS OR RADIO SHACK DISK BASIC, SPECIFY WHAT DISK BASIC.	$139.95
RADIO SHACK DISK CONTROLLER 1.1	$134.95

DISK DRIVE CABLES

CABLE FOR ONE DRIVE	$ 19.95
CABLE FOR TWO DRIVES	$ 24.95

MISC

64K UPGRADE W/MOD. INSTRUCTIONS, C,D,E,F, AND COCO 2	$ 49.95
HJL KEYBOARDS	$ 69.95
MICRO TECH LOWER CASE ROM ADAPTER	$ 74.95
RADIO SHACK BASIC 1.2	$ 29.95
RADIO SHACK DISK BASIC 1.1	$ 29.95
RADIO SHACK EXT. BASIC	$ 39.95
SCREEN CLEAN CLEARS UP VIDEO DISTORTION	$ 39.95
MEMOREX DISKS 5" SS,DD	$ 24.00
SHIPPING INCLUDED ON DISK PRICES	
DISK DRIVE CABINET & POWER SUPPLY	$ 49.95
SINGLE SIDED, DOUBLE DENSITY 5" DISK DRIVE	$199.95
DOUBLE SIDED, DOUBLE DENSITY 5" DISK DRIVE	$249.95

PRINTERS

EPSON RX-80	$325.00
EPSON RX-80FT	$375.00
EPSON MX-100	$650.00
EPSON FX-100	$799.00
EPSON FX-80	$549.00
EPSON MX-70	$200.00

SERIAL BOARDS FOR PRINTERS

MX-SERIES	$119.95
FX-SERIES	$ 99.99

USA ADD 2% SHIPPING
FOREIGN ADD 5% SHIPPING

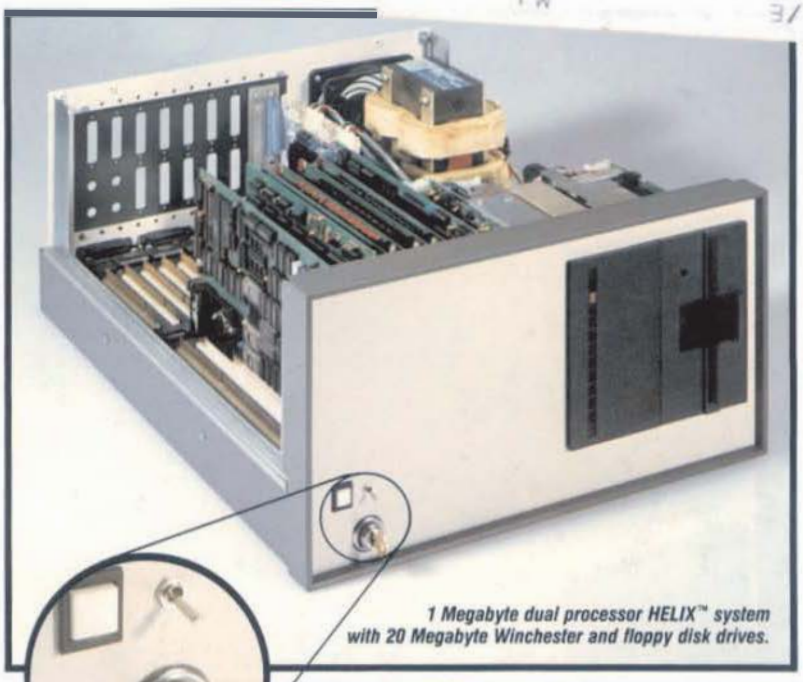
*FLEX is a Trademark of Technical System Consultants
*STAR DOS + is a Trademark of STAR-Kits & Data-Comp

5900 Cassandra Smith Rd. Hixson, TN 37343

68 Micro Journal
5900 Cassandra Smith Rd
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

MJ
MR. MICKEY FERGUSON
P. O. BOX 87
KINGSTON SPRINGS TN 37082
000422 A/E



1 Megabyte dual processor HELIX™ system with 20 Megabyte Winchester and floppy disk drives.

***ELWOOD COMPUTER SYSTEMS** leadership in computer technology by delivering the only computer system capable of switching between either the 6809 or the 68000 processor. Switching is easily accomplished by a simple front panel toggle switch. The reason we can offer this exclusive feature now, is that when our proven 6809 processor board was designed several years ago, we had the foresight to include the bus controls that allow processor switching.*

Hazelwood Computer Systems is also proud to be the first S-50/S-64 bus manufacturer to license and deliver the OS9/68K Operating System from Microware Systems Corporation. OS9/68K is the 68000 version of the popular and powerful OS9 Operating System. Utilizing our proven MC-20 disk controller, OS9/68K can conveniently share a Winchester disk with OS9. Changing from 6809 to 68000 operation is as simple as switching processors and booting the new system from the Winchester disk.

The ease of switching processors and operating systems makes a HELIX™ dual processor system the natural choice for software development. In addition, the advanced design of HELIX™ equipment, emphasizing performance and reliability, makes HELIX™ boards and systems the best value in computing offered anywhere.

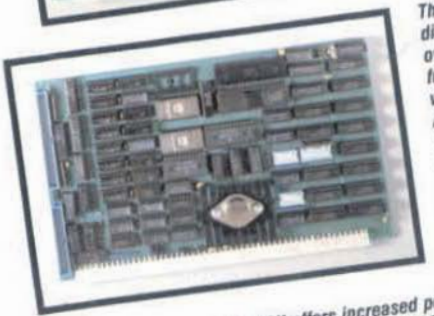
System prices vary with configuration. Call for exact pricing.

THE SWITCH IS ON...



The CP-08 processor board utilizes a 68008 processor running at 10 Mhz clock rate. Using proprietary bus synchronization circuitry and single cycle DMA, the CP-08 achieves a marked performance increase over a 2 MHz 6809. Offering absolute compatibility with the 68000 instruction set, the 68008 addresses up to 1 Megabyte of memory. Also included on the CP-08 are up to 4K of ROM, an interrupt timer, and, with battery backup operation, a clock/calendar and 2K RAM. Implemented as a standard S-50 board, the CP-08 brings 68000 operation to S-50 bus computers.

ORDER: CP-08 **PRICE: $595**



The MC-20 Mass Storage Controller board interfaces up to 4 floppy and 8 Winchester disk drives to the S-50/S-64 bus. The MC-20 is an intelligent controller with its own 2 Mhz 6809 processor and 56K RAM. It provides DMA data transfers to a full 24 bit address. All disk operation requests are by logical block number, with the controller performing the necessary track/sector address calculations. Any combination of 5¼ or 8 inch floppy drives can be accommodated with all drive parameters, such as write precompensation, software controlled for each individual drive. Winchester drives are connected via a SASI bus interface. Block address mapping is provided which allows a single drive to be segmented into several logical units. The MC-20 is the controller of the MS-20 Mass Storage Subsystem which includes a 20 Megabyte Winchester drive.

ORDER: MC-20 **PRICE: $695**

OS9/68K

OS9/68K offers increased performance and larger user memory space while retaining all of the features of OS9. Disk file compatibility and operational similarity assures that present OS9 users can easily transfer their operations to the 68000. Included are an editor, assembler, linker, and debugger. A C compiler is available now. BASIC09 and other languages will be avilable soon.

ORDER: OS9/68K **PRICE: $250**

All items available stock to 30 days.
Prices subject to change without notice.

HAZELWOOD COMPUTER SYSTEMS

907 East Terra, O'Fallon, MO 63366, 314-281-1055

HELIX™

OS9 and OS9/68K are registered trademarks of microware Systems Corp. HELIX is a trademark of Hazelwood Computer Systems.